arXiv:1211.3093v1 [cs.CG] 13 Nov 2012

Polynomial-time computation of homotopy groups and Postnikov systems in fixed dimension*

Martin Čadek[a] Marek Krčál[b] Jiří Matoušek[b,c] Lukáš Vokřínek[a]
Uli Wagner[d]

November 14, 2012

Abstract

For several computational problems in homotopy theory, we obtain algorithms with running time polynomial in the input size. In particular, for every *fixed* $k \geq 2$, there is a polynomial-time algorithm that, for a 1-connected topological space X given as a finite simplicial complex, or more generally, as a simplicial set with polynomial-time homology, computes the kth homotopy group $\pi_k(X)$, as well as the first k stages of a Postnikov system of X. Combined with results of an earlier paper, this yields a polynomial-time computation of $[X, Y]$, i.e., all homotopy classes of continuous mappings $X \to Y$, under the assumption that Y is $(k-1)$-connected and $\dim X \leq 2k-2$. We also obtain a polynomial-time solution of the *extension problem*, where the input consists of finite simplicial complexes X, Y, where Y is $(k-1)$-connected and $\dim X \leq 2k-1$, plus a subspace $A \subseteq X$ and a (simplicial) map $f\colon A \to Y$, and the question is the extendability of f to all of X.

The algorithms are based on the notion of a *simplicial set with polynomial-time homology*, which is an enhancement of the notion of a simplicial set with effective homology developed earlier by Sergeraert and his co-workers. Our polynomial-time algorithms are obtained by showing that simplicial sets with polynomial-time homology are closed under various operations, most notably, Cartesian products, twisted Cartesian products, and classifying space. One of the key components is also polynomial-time homology for the Eilenberg–MacLane space $K(\mathbb{Z}, 1)$, provided in another recent paper by Krčál, Matoušek, and Sergeraert.

*This research was supported by the ERC Advanced Grant No. 267165. The research of M. Č. and L. V. was supported by the project CZ.1.07/2.3.00/20.0003 of the Operational Programme Education for Competitiveness of the Ministry of Education, Youth and Sports of the Czech Republic. The research by M. K. was supported by the Center of Excellence – Inst. for Theor. Comput. Sci., Prague (project P202/12/G061 of GA ČR). The research by U. W. was supported by the Swiss National Science Foundation (grants SNSF-200020-138230 and SNSF-PP00P2-138948).

[a]Department of Mathematics and Statistics, Masaryk University, Kotlářská 2, 611 37 Brno, Czech Republic

[b]Department of Applied Mathematics, Charles University, Malostranské nám. 25, 118 00 Praha 1, Czech Republic

[c]Institute of Theoretical Computer Science, ETH Zurich, 8092 Zurich, Switzerland

[d]Institut de Mathématiques de Géométrie et Applications, École Polytechnique Fédérale de Lausanne, EPFL SB MATHGEOM, MA C1 553, Station 8, 1015 Lausanne, Switzerland

1 Introduction

One of the central themes in algebraic topology is understanding the structure of all continuous maps $X \to Y$, for given topological spaces X and Y (all maps between topological spaces in this paper are assumed to be continuous). Here two maps $f, g\colon X \to Y$ are usually considered equivalent if they are *homotopic*[1]; thus, the object of interest is $[X, Y]$, the set of all homotopy classes of maps $X \to Y$.

Computing higher homotopy groups. Many of the celebrated results throughout the history of topology can be cast as information about $[X, Y]$ for certain spaces X and Y. In particular, one of the important challenges propelling the research in algebraic topology has been the computation of the *homotopy groups of spheres*[2] $\pi_k(S^n)$, where only partial results have been obtained in spite of an enormous effort (see, e.g., [24, 17]).

Our concern here is the (theoretical) *complexity* of computing homotopy groups $\pi_k(Y)$ for an arbitrary Y. It is well known that the fundamental group $\pi_1(Y)$ is uncomputable, as follows from undecidability of the word problem in groups [23]. On the other hand, given a 1-connected space Y (i.e., one with $\pi_1(Y)$ trivial), say represented as a finite simplicial complex, there are algorithms that compute the higher homotopy group $\pi_k(Y)$, for every given $k \geq 2$. The first such algorithm is due to Brown [5], and newer ones have been obtained as a part of general computational frameworks in algebraic topology due to Schön [37], Smith [41], and Sergeraert and his co-workers (e.g., [38, 33, 29, 34]). In particular, an algorithm based on the methods of Sergeraert et al. can be found in Real [26]. We also refer to Romero and Sergeraert [30] for a new approach to homotopy computations.

The computation of the higher homotopy groups is generally considered very hard. The running time for the algorithms mentioned above has apparently never been analyzed. It is clear, however, that Brown's algorithm, which for a long time had been the only explicitly published algorithm for computing $\pi_k(Y)$, is heavily superexponential and totally unsuitable for actual computations.

Moreover, Anick [2] proved that computing $\pi_k(Y)$ is #P-hard,[3] where Y can even be assumed to be a 4-dimensional space, but, crucially, k is regarded as a part of the input. Actually, the hardness already applies to the potentially easier problem of computing the *rational homotopy groups* $\pi_k(Y) \otimes \mathbb{Q}$; practically speaking, one asks only for the rank of $\pi_k(Y)$, i.e., the number of direct summands isomorphic to $\mathbb{Z}$.

Anick's #P-hardness result has a caveat: it assumes Y to be given as a cell complex with a certain very compact representation. However, recently it was shown by the present

---

[1]Homotopy means a continuous deformation of one map into another. More precisely, f and g are defined to be homotopic, in symbols $f \sim g$, if there is a continuous $F\colon X \times [0,1] \to Y$ such that $F(\cdot, 0) = f$ and $F(\cdot, 1) = g$. With this notation, $[X, Y] = \{[f] : f\colon X \to Y\}$, where $[f] = \{g : g \sim f\}$ is the *homotopy class* of f.

[2]The kth homotopy group $\pi_k(Y)$ of a space Y is defined as the set of all homotopy classes of *pointed* maps $f\colon S^k \to Y$, i.e., maps f that send a distinguished point $s_0 \in S^k$ to a distinguished point $y_0 \in Y$ (and the homotopies F also satisfy $F(s_0, t) = y_0$ for all $t \in [0, 1]$). Strictly speaking, one should really write $\pi_k(Y, y_0)$ but for a path-connected Y, the choice of y_0 does not matter. Moreover, for 1-connected Y the pointedness of the maps does not matter either and one can identify $\pi_k(Y)$ with $[S^k, Y]$. Each $\pi_k(Y)$ is a group, which for $k \geq 2$ is Abelian, but the definition of the group operation is not important for us at the moment.

[3]Somewhat informally, the class of #P-hard problems consists of computational problems that should return a natural number (as opposed to YES/NO problems) and are at least as hard as counting the number of all Hamiltonian cycles in a given graph, or counting the number of subsets with zero sum for a given set of integers, etc. These problems are clearly at least as hard as NP-complete problems, and most likely even less tractable.

authors [7] that the computation of $\pi_k(Y)$ remains #P-hard even for a 4-dimensional simplicial complex Y, still with k a part of the input.

In view of the above, we consider the following result somewhat surprising.

Theorem 1.1. *For every fixed $k \geq 2$, there is a polynomial-time algorithm that, given a 1-connected space Y represented as a finite simplicial complex, or more generally, as a* simplicial set with polynomial-time homology *(this notion will be defined in Section 2), computes (the isomorphism type of) the kth homotopy group $\pi_k(Y)$.*

Here and in the sequel, the size of a simplicial complex is the number of simplices.

Since, under the conditions of the theorem, $\pi_k(Y)$ is a finitely generated Abelian group, it can be represented as a direct sum of finitely many cyclic groups, and the algorithm returns such a representation.

Let us remark that the algorithm does not need any certificate of the 1-connectedness of Y, but if Y is not 1-connected, the result may be wrong.

We should also mention that, although the theorem asserts the existence of an algorithm for every $k \geq 2$, we will actually present a single algorithm that accepts Y and k as input and outputs $\pi_k(Y)$, and for every k the running time is bounded by a polynomial in the size of Y, where the polynomial generally depends on k. However, for this setting, a single algorithm accepting Y and k, some of the formulations in the sequel would become more cumbersome, and so in the interest of simpler presentation, we stick to the setting as in Theorem 1.1. A similar remark applies to all of the other results below.

Remark: simple spaces. It can be checked that Theorem 1.1, as well as Theorem 1.2 below, hold, without any significant change in the proofs, under the assumtion that Y is a *simple* space (instead of 1-connected). This, somewhat technical, notion means that the fundamental group $\pi_1(Y)$ is possibly nontrivial but Abelian, and its action on each $\pi_k(Y)$, $k \geq 2$, is trivial. Here the action basically means "pulling the basepoint in Y along a loop"—see [15, pp. 341–342] for discussion. A natural example of simple spaces are *H-spaces*, which are a generalization of topological groups. In the interest of easier presentation we stick to the 1-connectedness assumption, though.

Computing Postnikov systems. The algorithm for computing $\pi_k(Y)$ in Theorem 1.1 is a by-product of a polynomial-time algorithm for computing the first k stages of a (standard) *Postnikov system* for a given space Y. In this respect it is similar to the algorithm of Brown [5] and some others, while, e.g., the algorithm in Real [26] is, in a sense, dual, building a *Whitehead tower* of Y. We note that with the tools used in the present paper, the Whitehead tower algorithm, too, could serve to prove Theorem 1.1.

A Postnikov system of a space Y is, roughly speaking, a way of building Y from "canonical pieces", called *Eilenberg–MacLane spaces*, whose homotopy structure is the simplest possible. A Postnikov system has countably many *stages* $P_0, P_1, \ldots$, where P_k reflects the homotopy properties of Y up to dimension k, and in particular, $\pi_i(P_k) \cong \pi_i(Y)$ for all $i \leq k$, while $\pi_i(P_k) = 0$ for $i > k$. The isomorphisms of the homotopy groups for $i \leq k$ are induced by maps $\varphi_i\colon Y \to P_k$, which are also a part of the Postnikov system. Moreover, there is a mapping $\mathbf{k}_i$ defined on P_i, called the ith *Postnikov class*; together with the group $\pi_{i+1}(Y)$ it describes how P_{i+1} is obtained from P_i, and it is of fundamental importance for dealing with maps from a space X into Y. We will say more about Postnikov systems later on; now we state the result somewhat informally.

Theorem 1.2 (informal)**.** *For every fixed $k \geq 2$, given a 1-connected space Y represented as a finite simplicial complex, or more generally, as a simplicial set with polynomial-time homology, a suitable representation of the first k stages of a Postnikov system for Y can be constructed, in such a way that each of the mappings φ_i and $\mathbf{k}_i$, $i \leq k$, can be evaluated in polynomial time.*

A precise statement will be given as Theorem 4.1.

Computing the structure of all maps. In the earlier paper [6] we provided an algorithm that, given finite simplicial complexes X and Y, computes the structure of $[X, Y]$ under the assumption that, for some $k \geq 2$, we have $\dim X \leq 2k - 2$ and Y is $(k-1)$-connected.[4] More precisely, under these assumptions, $[X, Y]$ has a canonical structure of a finitely generated Abelian group, and the algorithm determines its isomorphism type.

In the algorithm, the stage P_{2k-2} of the Postnikov system of Y is used as an approximation to Y, since for every 1-connected Y and every X of dimension at most $2k - 2$, there is an isomorphism $[X, Y] \cong [X, P_{2k-2}]$, induced by the composition with the mapping $\varphi_{2k-2} \colon Y \to P_{2k-2}$. At the same time, the continuous maps $X \to P_{2k-2}$ are easier to handle than the maps $X \to Y$: each of them is homotopic to a simplicial, and thus combinatorially described, map, and it is possible to define (and implement) a binary operation on P_{2k-2} which induces the group structure on $[X, P_{2k-2}]$. This, in a nutshell, explains the usefulness of the Postnikov system for dealing with maps into Y.

It is easy to check that the algorithm in [6] works in polynomial time in the size (number of simplices) of X and Y for every *fixed* k, provided that the first $2k-2$ stages of a Postnikov system for Y can be computed in polynomial time, as in Theorem 1.2 (the precise requirements on what should be computed can be found in [6]). We thus obtain the following result, anticipated in [6].

Corollary 1.3 (based on [6])**.** *For every fixed $k \geq 2$, there is a polynomial-time algorithm that, given finite simplicial complexes X, Y, where $\dim(X) \leq 2k - 2$ and Y is $(k-1)$-connected, computes the isomorphism type of $[X, Y]$ as an Abelian group. More generally, X can be a finite simplicial set and Y a simplicial set with polynomial-time homology.*

We will not dwell on the proof here, since it follows immediately by plugging the Postnikov system algorithm of Theorem 1.2 into the algorithm of [6] as a subroutine. We only remark that while the result of [6] is formulated for Y a finite simplicial complex, Y actually enters the computation solely through its Postnikov system, and so any Y can be handled for which the appropriate stages of the Postnikov system are available.

Computing extensions of maps. Related to the problem of determining $[X, Y]$ is the *extension problem*: given spaces A, X, Y, where $A \subseteq X$, and a map $f \colon A \to Y$, can f be extended to a map $X \to Y$? This is one of the most basic questions in algebraic topology, and a number of topological concepts, which may look quite advanced and esoteric to a newcomer, such as *Steenrod squares*, have a natural motivation in an attempt at a stepwise solution of the extension problem; see, e.g., Steenrod [43].

For $A \subseteq X$ and $f \colon A \to Y$ as above, let $[X, Y]_f \subseteq [X, Y]$ denote the set of all homotopy classes of maps $X \to Y$ that contain a map extending f.

One may also want to study the set of all extensions $\bar{f}$ of f with a finer equivalence relation than the ordinary homotopy of maps $X \to Y$, namely, homotopy fixing the map on A (i.e.,

[4]This means that $\pi_i(Y) = 0$ for all $i = 0, 1, \ldots, k-1$; a basic example is $Y = S^k$.

$\bar{f}_1, \bar{f}_2 \colon X \to Y$ are equivalent if they are connected by a homotopy $F \colon X \times [0,1] \to Y$ with $F(x,t) = f(x)$ for all $x \in A$ and $t \in [0,1]$). In order to distinguish these two notions, we refer to determining the structure of all extensions modulo homotopy fixing f on A as the *fine classification of the extensions of* f, and to determining $[X,Y]_f$ as the *coarse classification of the extensions of* f.

As a simple consequence of Theorem 1.2 and the methods of [6], we obtain the following.

Theorem 1.4 (Extendability of maps)**.** *Let $k \geq 2$ be fixed. Then there is a polynomial-time algorithm that, given finite simplicial complexes X, Y, a subcomplex $A \subseteq X$, and a simplicial map $f \colon A \to Y$, where $\dim(X) \leq 2k-1$ and Y is $(k-1)$-connected, decides whether f admits an extension to a (not neccessarily simplicial) map $X \to Y$.*

Moreover, if the extension exists and $\dim X \leq 2k-2$, the algorithm computes the structure of $[X,Y]_f$ as a coset in the Abelian group $[X,Y]$.

More generally, X can be a finite simplicial set and Y a simplicial set with polynomial-time homology.

The proof, assuming some of the material from [6], is presented in Section 5 below. We stress that, while f is given as a simplicial map (so that it can be specified by finite means), the extensions are considered as *arbitrary continuous maps*, and in particular, they are not assumed to be simplicial maps $X \to Y$.

Theorem 1.4 provides a coarse classification of all extension assuming $\dim X \leq 2k-2$. We have also worked out an algorithm that, under the same conditions, provides a fine classification of all extensions, but the details are more demanding and we defer them to a future paper.

For the next higher dimension $\dim X = 2k-1$, although the existence of an extension can be decided, we can no longer produce the coarse classification of all extensions, and we suspect that this problem should be intractable in a suitable sense.

The assumption on X and Y in Corollary 1.3 may perhaps look artificial at first sight. However, it is needed for $[X,Y]$ to have a canonical structure of an Abelian group. Moreover, the similar assumption in Theorem 1.4 (with $\dim X$ one higher) turns out to be sharp, in the following sense: In [7] we show that the extendability problem is *algorithmically undecidable* for finite simplicial complexes $A \subseteq X$ and Y and a simplicial map $f \colon A \to Y$ with $\dim X = 2k$ and $(k-1)$-connected Y. Moreover, for every $k \geq 2$, there is a *fixed* $(k-1)$-connected $Y = Y_k$ such that the extension problem for maps into Y_k, with A, X, f as the input, $\dim X \leq 2k$, is undecidable. In a similar sense, $X = X_k$ and $A = A_k$ can be fixed, so that the input consists only of Y and f, and undecidability still holds. See [7] for more details.

Methods. The results of this paper rely on a number of known methods and techniques. We see the main contributions in selecting suitable methods among various available alternatives and adapting them for our purposes, assembling everything together, and setting up a framework for dealing with polynomial-time algorithms of a somewhat unusual kind.

We have also made a significant effort to present the results in an accessible manner. The required techniques involve a large amount of material, and methods from two traditionally separated areas, algebraic topology and algorithm design, need to be brought together. We expect the number of potential readers moving with ease in both of these areas to be rather small, and thus we try to make the exposition as self-contained as reasonably possible, sometimes covering things which may be considered well known in one of the areas.

The Postnikov system algorithm, on the top level, essentially follows the approach of Brown [5] (we have examined proofs of existence of a Postnikov system in standard textbooks,

such as [15, 42], and none seemed quite suitable for our purposes). But Brown's algorithm in the original form uses a straightforward representation of simplicial Eilenberg–MacLane spaces, and thus it works only for input spaces with all the relevant homotopy groups finite. In the case of infinite homotopy groups, the corresponding Eilenberg–MacLane spaces are simplicial sets with *infinitely many* nondegenerate simplices in the relevant dimensions. For dealing with these, and with other infinitary objects derived from them in the course of the algorithm, we follow the paradigm of *objects with effective homology* developed by Sergeraert, Rubio, Real, Dousson, and Romero (see, e.g., [38, 33, 29, 34]; the lecture notes [35] provide the most detailed exposition available so far). Some of their results have never appeared in peer-reviewed journals; for example, for some of the operations needed in the present paper, we use methods described in some detail, as far as we know, only in Real's PhD. thesis [25] written in Spanish.

For the purpose of polynomial-time computations, we replace effective homology with *polynomial-time homology*, as introduced in [18]. Thus, we need polynomial-time versions of all the required operations in effective homology.

There is one case, namely, polynomial-time homology for the simplicial Eilenberg–MacLane space $K(\mathbb{Z}, 1)$, where we had to develop a new algorithm, since the classical one is not polynomial in general. This part is not provided here, but rather in the companion paper [18]; the methods used in that paper have flavor somewhat distinct from those employed here, and we feel that a combined paper would be too extensive and cumbersome.

In all other cases, we could rely on known algorithms. Verifying their polynomiality sometimes still requires nontrivial analysis and assumptions. Moreover, since the intermediate objects used in the algorithms are of somewhat unusual kind from the computer science point of view, we need to set up a suitable formal framework in order to make claims about polynomial running time.

Fixed-parameter tractability? An interesting question is the dependence of the running time in the theorems above on the fixed parameter k. As indicated by the #P-hardness result alluded to above, polynomial dependence on k is extremely unlikely. However, one may ask whether some of the considered computational problems are *fixed-parameter tractable*,[5] or rather W[1]-hard or even harder.

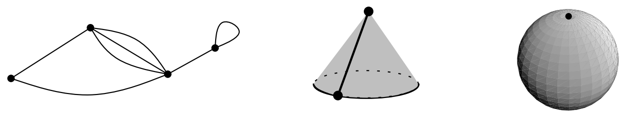
2 Simplicial sets and chain complexes with polynomial-time homology

2.1 Preliminaries on simplicial sets and chain complexes

Simplicial sets. A simplicial set is a way of specifying a topological space in purely combinatorial terms; we can think of it as an instruction manual telling us how the considered space should be assembled from simple building blocks. All topological spaces in the considered algorithms are going to be represented in this way. Simplicial sets can be regarded as a generalization of simplicial complexes; they are formally more complicated, but more powerful and flexible. We refer to [12, 39] for an introduction, to [8, 19] as compact more comprehensive sources, and to [13] for a more modern treatment.

[5]A computational problem involving a parameter k is *fixed-parameter tractable* if it admits an algorithm with running time bounded by $f(k)n^C$, where n measures the input size, $f(k)$ is an arbitrary function of k, and C is a constant *independent* of k; see, e.g., [22] for an introduction to *parameterized complexity*.

Similar to a simplicial complex, a simplicial set is a space built of vertices, edges, triangles, and higher-dimensional simplices, but simplices are allowed to be glued to each other and to themselves in more general ways. For example, one may have several 1-dimensional simplices connecting the same pair of vertices, a 1-simplex forming a loop, two edges of a 2-simplex identified to create a cone, or the boundary of a 2-simplex all contracted to a single vertex, forming an S^2.



Another new feature of a simplicial set, in comparison with a simplicial complex, is the presence of *degenerate simplices.* For example, the edges of the triangle with a contracted boundary (in the last example above) do not disappear, but each of them becomes a degenerate 1-simplex.

A simplicial set X is represented as a sequence $(X_0, X_1, X_2, \ldots)$ of mutually disjoint sets, where the elements of X_k are called the *k-simplices of* X (we note that, unlike for simplicial complexes, a simplex in a simplicial set need not be determined by the set of its vertices; indeed, there can be many simplices with the same vertex set). For every $k \geq 1$, there are $k+1$ mappings $\partial_0, \ldots, \partial_k \colon X_k \to X_{k-1}$ called *face operators*; the intuitive meaning is that for a simplex $\sigma \in X_k$, $\partial_i \sigma$ is the face of σ opposite to the ith vertex. Moreover, there are $k+1$ mappings $s_0, \ldots, s_k \colon X_k \to X_{k+1}$ (opposite direction) called the *degeneracy operators*; the approximate meaning of $s_i \sigma$ is the degenerate simplex which is geometrically identical to σ, but with the ith vertex duplicated.

A simplex is called *degenerate* if it lies in the image of some s_i; otherwise, it is *nondegenerate.* We write X^{ndg} for the set of all nondegenerate simplices of X. We call X *finite* if X^{ndg} is finite (every nonempty simplicial set has infinitely many degenerate simplices).

There are natural axioms that the ∂_i and the s_i have to satisfy, but we will not list them here, since we will not really use them. Moreover, the usual definition of simplicial sets uses the language of category theory and is very elegant and concise; see, e.g., [13, Section I.1].

Every simplicial set X specifies a topological space $|X|$, the *geometric realization* of X. It is obtained by assigning a geometric k-dimensional simplex to each nondegenerate k-simplex of X, and then gluing these simplices together according to the face operators; we refer to the literature for the precise definition.

Simplicial maps. For simplicial sets X, Y, a *simplicial map* $f \colon X \to Y$ is a sequence $(f_k)_{k=0}^{\infty}$ of maps $f_k \colon X_k \to Y_k$ (every k-simplex is mapped to a k-simplex) that commute with the face and degeneracy operators, i.e., $\partial_i f_k = f_{k-1} \partial_i$ and $s_i f_k = f_{k+1} s_i$. We let $\mathrm{SMap}(X, Y)$ stand for the set of all simplicial maps $X \to Y$.

It is useful to observe that it suffices to specify a simplicial map $f \colon X \to Y$ on the *nondegenerate* simplices of X; the values on the degenerate simplices are then determined uniquely. In particular, if X is finite, then such an f can be specified as a finite object.

Every simplicial map $f \colon X \to Y$ defines a continuous map $\varphi \colon |X| \to |Y|$ of the geometric realizations. There is a very important class of simplicial sets, called *Kan simplicial sets*, with the following crucial property: if Y is a Kan simplicial set and X is any simplicial set, then every continuous map $\varphi : |X| \to |Y|$ is homotopic to (the geometric realization of) some simplicial map $f \colon X \to Y$. This is essential in the algorithmic treatment of continuous maps. Here we omit the definition of a Kan simplicial set, since we will not directly use it.

Chain complexes. Together with a simplicial set X, we will consider its associated *normalized chain complex* $C_*(X)$, but sometimes the algorithms will also need other types of chain complexes.

For our purposes, it is sufficient to use the kind of chain complexes usually considered in introductory textbooks when defining homology and cohomology groups. Thus, in the sequel, a *chain complex* C_* is a sequence $(C_k)_{k\in\mathbb{Z}}$ of *free* Abelian groups (in other words, free $\mathbb{Z}$-modules), together with a sequence $(d_k\colon C_k \to C_{k-1})_{k\in\mathbb{Z}}^{\infty}$ of group homomorphisms that satisfy the condition $d_{k-1}d_k = 0$.[6] The C_k are the *chain groups*, their elements are called *k-chains*, and the d_k the *differentials*. If c is a k-chain, we sometimes say that the *degree* of c equals k. We will work only with chain complexes C_* with $C_k = 0$ for all $k < 0$.

We also recall that $Z_k = Z_k(C_*) := \ker d_k \subseteq C_k$ is the group of *cycles*, $B_k = B_k(C_*) := \operatorname{im} d_{k+1} \subseteq Z_k$ is the group of *boundaries*, and the quotient group $H_k(C_*) := Z_k/B_k$ is the kth *homology group* of the chain complex C_*.

For the normalized chain complex $C_*(X)$ of a simplicial set X mentioned above, the kth chain group $C_k(X)$ is the free Abelian group over X_k^{ndg}, the set of all k-dimensional *nondegenerate* simplices (in particular, $C_k(X) = 0$ for $k < 0$).[7] This means that a k-chain is a formal sum

$$c = \sum_{\sigma\in X_k^{\mathrm{ndg}}} \alpha_\sigma \cdot \sigma,$$

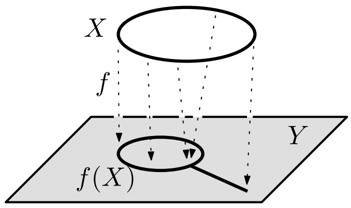
where the α_σ are integers, only finitely many of them nonzero. The differentials are defined in a standard way using the face operators: for k-chains of the form $1\cdot\sigma$, which constitute a basis of $C_k(X)$, we set $d_k(1\cdot\sigma) := \sum_{i=0}^{k}(-1)^i \cdot \partial_i\sigma$ (some of the $\partial_i\sigma$ may be degenerate simplices; then they are ignored in the sum), and this extends to a homomorphism in a unique way ("linearly").

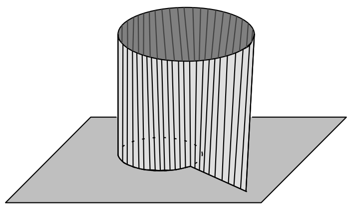
Let C_* and $\tilde{C}_*$ be two chain complexes. We recall that a *chain map* $f\colon C_* \to \tilde{C}_*$ is a sequence $(f_k)_{k\in\mathbb{Z}}$ of homomorphisms, $f_k\colon C_k \to \tilde{C}_k$, compatible with the differentials, i.e., $f_{k-1}d_k = \tilde{d}_k f_k$. A simplicial map $f\colon X \to Y$ of simplicial sets induces a chain map $f_*\colon C_*(X) \to C_*(Y)$ in the obvious way.

Mapping cylinder and mapping cone. We recall two standard constructions for topological spaces, and then we mention their algebraic counterparts. Let $f\colon X \to Y$ be a map of topological spaces. Then the *mapping cylinder* $\mathrm{Cyl}(f)$ is obtained by gluing the product ("cylinder") $X \times [0,1]$ to Y via the identification of $(x,0)$ with $f(x) \in Y$, for all $x \in X$, as the next picture indicates.

[6] More generally, instead of $\mathbb{Z}$-modules, one might consider modules over a commutative ring R, and they need not be free. Moreover, in the literature, the operations considered in Section 3 below are sometimes presented in a still more general algebraic setting, with *differential modules* replacing chain complexes. Here we prefer the more concrete setting of chain complexes, mainly in order to avoid burdening the presentation with additional notions.

[7] In the literature, the notation $C_*(X)$ is sometimes used for another chain complex associated with X, where the degenerate simplices also appear as generators (it yields the same homology as the normalized chain complex). But since we will work exclusively with the normalized chain complex, we reserve the simple notation $C_*(X)$ for these.





Cyl(f)

The mapping cone Cone(f) is obtained from the mapping cylinder Cyl(f) by contracting the "top copy" of X, i.e., the subspace $X \times \{1\}$, to a single point.

We will not use these geometric constructions directly. In one of the proofs, we will need a simplicial version of the mapping cylinder, in a setting where X, Y are simplicial sets and f is a simplicial map, and we will introduce it at the appropriate moment. Otherwise, we will work exclusively with *algebraic analogs* of these constructions. Conceptually, they are obtained by considering how the chain complexes of Cyl(f) and Cone(f) are related to the chain complexes of X and Y and to the chain map f_* induced by f, and then generalizing to arbitrary chain complexes and chain map.

The resulting definitions are as follows. Let $C_*, \tilde{C}_*$ be chain complexes and let $\varphi\colon C_* \to \tilde{C}_*$ be a chain map. Then the (algebraic) *mapping cylinder* $\mathrm{Cyl}_*(\varphi)$ has chain groups $\mathrm{Cyl}_k := C_{k-1} \oplus \tilde{C}_k \oplus C_k$ (a direct sum), and the differential given by

$$d_k^{\mathrm{Cyl}_*(\varphi)}(a, \tilde{c}, b) := (-d_{k-1}(a), \varphi_k(a) + \tilde{d}_k(\tilde{c}), -a + d_k(b)),$$

where d is the differential of C_* and $\tilde{d}$ is the differential of $\tilde{C}_*$.

In a similar spirit, the (algebraic) *mapping cone* $\mathrm{Cone}_*(\varphi)$ of φ is the chain complex whose kth chain group is the direct sum $C_{k-1} \oplus \tilde{C}_k$, and with the differential given by

$$d_k^{\mathrm{Cone}_*(\varphi)}(a, \tilde{b}) := (-d_{k-1}(a), \varphi_k(a) + \tilde{d}_k(\tilde{b})), \quad (a, \tilde{b}) \in C_{k-1} \oplus \tilde{C}_k. \tag{1}$$

For later use, we note that the canonical inclusion $i\colon \tilde{C}_* \to \mathrm{Cone}_*(\varphi)$, given by $i(\tilde{b}) = (0, \tilde{b})$, is a chain map, as can be seen from (1); on the other hand, the other canonical inclusion $j\colon C_* \to \mathrm{Cone}_*(\varphi)$ is not a chain map (it does not respect degrees, and it does not commute with the face operators, unless $\varphi = 0$).

2.2 The meaning of "computing $\pi_{17}(Y)$ in polynomial time"

In computational complexity theory, which is a branch of computer science that focuses on classifying computational problems according to their inherent difficulty, algorithms are usually represented as Turing machines, or some other models of a general computing machine. Such an algorithm accepts an input $u \in \Sigma^*$, where Σ is some fixed finite alphabet (for our purposes, we may assume w.l.o.g. that $\Sigma = \{0, 1\}$ is the binary alphabet), and where Σ^* denotes the set of all strings (finite sequences) of symbols of Σ. Given $u \in \Sigma^*$, the algorithm computes some output $v \in \Sigma^*$.

We say that a mapping $f\colon \Sigma^* \to \Sigma^*$ is a *polynomial-time mapping* if there is an algorithm A and a polynomial $p(x)$ such that, for every input $u \in \Sigma^*$, the algorithm A outputs $f(u)$ after at most $p(|u|)$ steps, where $|u|$ denotes the length (number of symbols) of u.

It is easy to see that the composition of two polynomial-time mappings is again a polynomial-time mapping. (Here we use that if the computation of $f(u)$ takes at most $p(|u|)$ steps, then

$|f(u)| \le p(|u|)$, for otherwise, the algorithm for evaluating f would not have enough time to write $f(u)$ down.) We will frequently use this fact, often without mentioning it explicitly.

Encoding size. Thus, the notion of polynomial time is very straightforward, although not easy to study, for mappings assigning strings to strings. However, if we consider "real-life" computational tasks, such as testing whether a given natural number n is a prime, or computing $\pi_{17}(Y)$ for a simplicial complex Y, then neither the input nor the output are *a priori* strings. In order to talk about the computational complexity of such tasks, we first need to specify some *encoding* of the input and output objects by strings.

For testing primality, we thus need to specify an (injective) function $\mathsf{enc}\colon \mathbb{N} \to \Sigma^*$ assigning a string to every natural number (here we consider the encoding of the two possible outputs, YES and NO, as too trivial to discuss). The most usual choice is representing n by the standard binary notation; e.g., $\mathsf{enc}(17) = 10001$. In this paper we assume binary encoding of all integers (unless stated otherwise). With this encoding, the possibility of primality testing in polynomial time is a celebrated recent result. However, if we chose a different, *unary* encoding enc', which represents n by a string of n ones, e.g., $\mathsf{enc}'(17) = 11111111111111111$, then testing primality in polynomial time becomes very easy—we can afford to test all possible divisors from 2 to $n-1$. This example illustrates that sometimes the choice of encoding may be very significant.[8]

For discussing polynomial-time algorithms, we often do not need to specify the encoding function enc completely. Usually we suffice with the *encoding size*, where the size of an object a is $\mathsf{size}(a) = |\,\mathsf{enc}(a)|$, the number of bits in its encoding. In the above example with primality, we had $\mathsf{size}(n) = \lfloor 1 + \log_2 n \rfloor$ for the binary encoding and $\mathsf{size}'(n) = n$ for the unary encoding.

We note that changes in the encoding that transform the size by at most a fixed polynomial, e.g., replacing $\mathsf{size}(a)$ with $\mathsf{size}'(a) = (37\,\mathsf{size}(a) + 100)^{26}$, leave the notion of a polynomial-time mapping unchanged. Thus, for the purpose of developing polynomial-time algorithms, we usually need not describe the encoding in much detail.

The encoding size of simplicial complexes and of Abelian groups. We recall that a finite simplicial complex Y can be regarded as a hereditary system of subsets of a finite vertex set V (hereditary meaning that if $\sigma \in Y$ and $\sigma' \subseteq \sigma$, then $\sigma' \in Y$ as well). For encoding such an Y, we can identify V with the set $\{1, 2, \ldots, n\}$, and then represent Y as a list of all simplices, where each simplex is given by the list of vertices. Thus, if the dimension of Y is bounded by a constant (as we may assume in all of our results), $\mathsf{size}(Y)$ is bounded by a polynomial function of the number of simplices of Y, and so for the purpose of discussing polynomial-time algorithms, we may assume that $\mathsf{size}(Y)$ equals the number of simplices.

The elements of the homotopy group $\pi_{17}(Y)$ are, by definition, equivalence classes of pointed maps $S^{17} \to Y$, and it is far from obvious how even a single such element could be represented by a string. However, our algorithm computes only the *isomorphism type* of the homotopy groups. (Computing a reasonable representation for the mappings corresponding to the generators of the homotopy group is currently an interesting open problem.)

It is known that, for a finite simplicial complex Y and $k \ge 2$, $\pi_k(Y)$ is a finitely generated

[8] Here is another example, closer to our topics, of how the encoding may matter: If a simplicial complex K is given by a list of *all* of its simplices, as we are going to assume here, then computing the Euler characteristic $\chi(K)$ is a trivial matter and can obviously be done in polynomial time. However, if K is specified by listing only the *maximal* simplices, and if we do *not* assume $\dim K$ fixed, then computing $\chi(K)$ is #P-hard [31], and thus extremely unlikely to be polynomial-time solvable!

Abelian group; this actually also follows from the analysis of our algorithm. A well-known structure theorem asserts that each finitely generated Abelian group π can be represented as a direct sum $\mathbb{Z}^r \oplus (\mathbb{Z}/m_1) \oplus (\mathbb{Z}/m_2) \oplus \cdots \oplus (\mathbb{Z}/m_s)$ of cyclic groups.[9] We are going to encode it by the $(r+s)$-tuple $\mathbf{m} = (\underbrace{0, 0, \ldots, 0}_{r \text{ zeros}}, m_1, \ldots, m_s)$, where $m_1, \ldots, m_s$ are encoded in binary. Thus, we may take

$$\mathsf{size}(\pi) = r + \sum_{i=1}^{s} \mathsf{size}(m_i).$$

The reader may wonder why r is not encoded in binary as well. The reason is pragmatic; we will also be using finitely generated Abelian groups as inputs to certain auxiliary algorithms, and we would not be able to make these auxiliary algorithms polynomial with r encoded in binary. A heuristic explanation for this is that an *element* of $\mathbb{Z}^r$ is an r-tuple of integers, and thus an encoding of such an element has size at lest proportional to r. If the encoding size of $\mathbb{Z}^r$ were of order $\log_2 r$, then a polynomial-time algorithm working with $\mathbb{Z}^r$ would not be able even to read or write any single group element.

This specification of encoding sizes gives a precise meaning to the polynomiality claim in Theorem 1.1. We note that the polynomiality of the algorithm also implies the (non-obvious) claim that, for k fixed, $\mathsf{size}(\pi_k(Y))$ is bounded by a polynomial function of $\mathsf{size}(Y)$.

2.3 Locally polynomial-time simplicial sets and chain complexes

In what sense do we construct a Postnikov system? As was mentioned after Theorem 1.2, the stages $P_k = P_k(Y)$ of a Postnikov system of Y can be regarded as approximations of Y, which are in some sense easier to work with than Y itself. The price to pay is that even if Y is a finite simplicial complex, the P_k are simplicial sets that usually have infinitely many nondegenerate simplices in each dimension.

In many areas where computer scientists seek efficient algorithms, the algorithms work with finite objects, such as finite graphs or matrices, and there is no problem with explicitly representing such objects in the computer memory. This contrasts with the situation for the P_k, where we cannot produce the infinite list of all simplices of a given dimension explicitly. Thus, the question arises, in what sense we construct P_k and how we can work with it.

A complete answer is that we want to equip P_k with polynomial-time homology, which is a notion defined later. For now, we give at least a partial answer: We certainly want to be able to inspect *locally* every given piece of P_k. For example, for every fixed k and ℓ, given any ℓ-dimensional simplex σ of P_k, and an integer $i \in \{0, 1, \ldots \ell\}$, we should be able to compute the ith face $\partial_i \sigma$, the ith degeneracy $s_i \sigma$, and also the value $\mathbf{k}_k(\sigma)$ of the Postnikov class at σ. Because of the infinite domains, the mappings ∂_i, s_i, and $\mathbf{k}_k$ cannot be given by a finite table (somewhat exceptionally, the mapping $\varphi_k \colon Y \to P_k$ could be represented by a table if Y is finite). Instead, each of them is going to be given as an *algorithm*.

Thus, we are going to represent stage of the Postnikov system by a collection of algorithms, and similarly for various other infinite simplicial sets, chain complexes, and some other kinds of objects. In computer science, this is sometimes called a *black box* or *oracle* representation.[10]

[9] Moreover, we may assume that the m_i satisfy the divisibility condition $m_1 | m_2 | \cdots | m_s$, in which case these orders are determined uniquely from π and thus describe its isomorphism type.

[10] Professor Sergeraert has suggested an alternative framework, inspired by functional programming, for dealing with computational complexity of algorithms similar to those considered in the present paper. It

Polynomiality. Since we want to use the stages of the Postnikov system in polynomial-time algorithms, such as the one in Corollary 1.3 (the computation of $[X,Y]$), we obviously want that the black boxes representing $P_k = P_k(Y)$ work in polynomial time. But some care is needed in formulating such a requirement.

For example, let us consider the Postnikov class $\mathbf{k}_{17}$, which is a simplicial map from P_{17} into another simplicial set, namely, the Eilenberg–MacLane space $K(\pi_{18}(Y), 19)$, to be introduced in Section 3.7. The simplices of P_{17}, as well as those of $K(\pi_{18}(Y), 19)$, are canonically represented by certain ordered collections of integers (or sometimes elements of some $\mathbb{Z}/m$), and it might happen that while $\mathsf{size}(\sigma)$ is a constant, $\mathsf{size}(\mathbf{k}_{17}(\sigma))$ also depends on the input simplicial complex Y and becomes arbitrarily large for some choices of Y.[11] Then $\mathbf{k}_{17}(\sigma)$ cannot be evaluated in time polynomial in $\mathsf{size}(\sigma)$.

Even if, for every input Y, we could compute $\mathbf{k}_{17}(\sigma)$ in time polynomial in $\mathsf{size}(\sigma)$, it might happen that the polynomial depended on Y. For example, we might encounter a sequence $Y^{(1)}, Y^{(2)}, \ldots$ of inputs such that $\mathsf{size}(Y^{(j)}) \leq j$, say, and the time for evaluating $\mathbf{k}_{17}(\sigma)$ is $\mathsf{size}(\sigma)^j$. Then we would not be able to use such a Postnikov stage an algorithm such as the one for computing $[X,Y]$ (Corollary 1.3), where the running time should depend polynomially on $\mathsf{size}(Y)$.

Thus, we cannot simply require $\mathbf{k}_{17}(\sigma)$ to be computed in time polynomial in $\mathsf{size}(\sigma)$. Instead, we are going to require the running time to be bounded by a polynomial in $\mathsf{size}(Y) + \mathsf{size}(\sigma)$ (where the polynomial depends on $\dim \sigma$ and on k, the index of the Postnikov stage).

To get Y in the picture, we introduce *parameterized simplicial sets*; these are families of simplicial sets, typically with infinitely many members, where each member of the family is described by some value of a parameter. We assume some agreed-upon encoding of the parameters by strings, and the length of the encoding strings is taken as the size of the corresponding simplicial set in the family. Then we assume that the black boxes, such as the one for evaluating $\mathbf{k}_{17}$, take both the parameter value and σ as input, and that they run in time polynomial in the size of this combined input.

Locally polynomial-time simplicial sets. At this moment we postpone further discussion of the Postnikov stages P_k and the Postnikov classes $\mathbf{k}_k$ until Section 4, and we introduce a general notion of a simplicial set represented "locally" by polynomial-time black boxes.

Definition 2.1 (Locally polynomial-time simplicial set)**.** *Let $\mathcal{I}$ be a set, on which an injective mapping $\mathsf{enc}\colon \mathcal{I} \to \Sigma^*$ is defined, specifying an encoding of each element of $\mathcal{I}$ by a string; we will refer to $\mathcal{I}$ as a* parameter set*. We define a* parameterized simplicial set *as a mapping X that, for some parameter set $\mathcal{I}$, assigns to each $I \in \mathcal{I}$ a simplicial set $X(I)$. Sometimes we will write such a parameterized simplicial set as $(X(I) : I \in \mathcal{I})$. We also assume that an encoding of simplices by strings has been fixed for each of the simplicial sets $X(I)$.*

We say that such an X is a locally polynomial-time simplicial set *if, for each k, there is an algorithm that, given $I \in \mathcal{I}$, a k-dimensional simplex $\sigma \in X(I)_k$, and $i \in \{0, 1, \ldots, k\}$, computes $\partial_i \sigma$ in time polynomial in $\mathsf{size}(I) + \mathsf{size}(\sigma)$ (where the polynomial may depend on k), and there is a similar algorithm for evaluating the degeneracy operators $s_i \sigma$.*

should be presented in [36].

[11] Here is an example of a similar phenomenon in a simpler and perhaps more familiar setting. Suppose that we want to represent the elements of the cyclic group $\mathbb{Z}/m$ by the integers $0, 1, \ldots, m-1$, and we want an algorithm for computing the inverse element $-i$ for a given i. Then we cannot require the algorithm to run in time polynomial in $\mathsf{size}(i)$, because for $i = 1$ the output must be $m-1$, and its encoding size depends on m—at least if we use the standard binary encoding of the integers. A reasonable requirement is to bound the running time polynomially in $\mathsf{size}(m)$.

Let $(X(I) : I \in \mathcal{I})$ and $(Y(I) : I \in \mathcal{I})$ be parameterized simplicial sets with the same parameter set, and for each $I \in \mathcal{I}$, let f_I be a simplicial map $X(I) \to Y(I)$. We say that $f = (f_I\colon I \in \mathcal{I})$ is a polynomial-time simplicial map $X \to Y$ *if for each $k \geq 0$, there is an algorithm that, given $I \in \mathcal{I}$ and $\sigma \in X(I)_k$, computes $f_I(\sigma)$ in time polynomial in* $\mathsf{size}(I) + \mathsf{size}(\sigma)$.[12]

As was explained above, the main purpose of the parameterized setting is to make the polynomial bounds on the running time of the black boxes uniform in the input of the considered algorithms. Let us remark that for effective homology in the setting of Sergeraert et al. [35], where one only wants the existence of algorithms and does not analyze their running time, no uniformity and no parameterization is needed, and one can work with individual simplicial sets, each equipped with its own black boxes.

We will see numerous examples of locally polynomial-time simplicial sets later on. Of course, the Postnikov stage $P_k = P_k(Y)$, parameterized by the set of all finite 1-connected simplicial complexes, is going to be one such example. (However, P_k also has an additional structure besides being a locally polynomial-time simplicial set.)

Another, rather simple, example is made of all finite simplicial sets, as will be discussed at the end of the present section. Others can be built from this one by applying various operations, such as products or twisted products, which will be considered later.

Locally polynomial-time chain complexes. First, let $(X(I) : I \in \mathcal{I})$ be a locally polynomial-time simplicial set, and let $C_*(X(I))$ be the normalized chain complex of $X(I)$. This gives us a chain complex parameterized by $\mathcal{I}$. The k-chains of $C_*(X(I))$ are finite sums of the form $c = \sum_{\sigma:\alpha_\sigma\neq 0} \alpha_\sigma \cdot \sigma$, where the σ are nondegenerate simplices of $X(I)_k$, and we can represent such a c by a list of simplices and of the corresponding nonzero coefficients. Thus we naturally put $\mathsf{size}(c) := \sum_{\sigma:\alpha_\sigma\neq 0}(\mathsf{size}(\sigma) + \mathsf{size}(\alpha_\sigma))$.

For this representation, it is easy to check that the addition and subtraction of k-chains, as well as the differentials, can be computed in time polynomial in $\mathsf{size}(I)$ plus the size of the chains involved. (For this, we need to observe that, given a simplex $\sigma \in X(I)$, we can test whether it is degenerate, since every degenerate σ satisfies $\sigma = s_i\partial_i\sigma$ for some i.)

We will also need to work with chain complexes that are not necessarily normalized chain complexes of simplicial sets. We will need that the chain groups are "effectively free," meaning that the chains are represented by coefficients w.r.t. some fixed basis. The following definition is a direct analog of the definition of a locally polynomial-time simplicial set, and it includes the normalized chain complex of a locally polynomial-time simplicial set as a special case.

Definition 2.2 (Locally polynomial-time chain complex). *Let $\mathcal{I}$ be a parameter set as in Definition 2.1, and let $(C(I)_* : I \in \mathcal{I})$ be a parameterized chain complex, i.e., a mapping assigning a chain complex to each $I \in \mathcal{I}$. We say that such a parameterized chain complex is a* locally polynomial-time chain complex *if the following hold.*

(i) *For each $C(I)_*$ and each k, there is a basis* $\mathrm{Bas}_k = \mathrm{Bas}(I)_k$ *of $C(I)_k$ (possibly infinite), which we call the* distinguished basis[13] *of $C(I)_k$, and whose elements have some agreed-upon encoding by strings. An arbitrary k-chain $c \in C(I)_k$ is (uniquely) represented as*

[12] More generally, we might want to consider a simplicial map f_I that goes from $X(F(I))$ to $Y(G(I))$, for some polynomial-time computable maps F, G. By composing algorithms we may think of $X(F(I))$ and $Y(G(I))$ as simplicial sets parameterized by I and thus this seemingly more general notion can be interpreted as a polynomial-time simplicial map in our sense.

[13] Chain complexes with a distinguished basis for each chain group are sometimes called *cellular*.

an integer linear combination of elements of $\mathrm{Bas}(I)_k$*, i.e., by a finite list of elements of* $\mathrm{Bas}(I)_k$ *and the corresponding nonzero coefficients. (This also defines the encoding size for chains.)*

(ii) *For every fixed k, there is an algorithm for evaluating the differential* d_k *of* $C(I)_*$*, which computes* $d_k(c)$ *in time polynomial in* $\mathsf{size}(I) + \mathsf{size}(c)$.

We note that in the representation of k-chains as in (i), the chains $c + c'$ and $c - c'$ can be computed in time polynomial in $\mathsf{size}(c) + \mathsf{size}(c')$, even without including $\mathsf{size}(I)$.

If $(C(I)_* : I \in \mathcal{I})$ and $(\tilde{C}(I)_* : I \in \mathcal{I})$ are parameterized chain complexes, then, in complete analogy with polynomial-time simplicial maps in Definition 2.1, we define a *polynomial-time chain map* $\varphi = (\varphi_I)_{I\in\mathcal{I}} \colon C_* \to \tilde{C}_*$, where each φ_I is a chain map $C(I)_* \to \tilde{C}(I)_*$, such that for each fixed k, $(\varphi_I)_k(c)$ can be computed in time polynomial in $\mathsf{size}(I) + \mathsf{size}(c)$.

Changing the parameter or: preprocessing. Let $(X(J) : J \in \mathcal{J})$ be a parameterized simplicial set, and let $F \colon \mathcal{I} \to \mathcal{J}$ be a polynomial-time mapping of another parameter set $\mathcal{I}$ into $\mathcal{J}$. Then we can define a new parameterized simplicial set $(\tilde{X}(I) : I \in \mathcal{I})$ by $\tilde{X}(I) := X(F(I))$; if X is locally polynomial-time, then so is $\tilde{X}$.

In our algorithms, X can often be regarded as a version of $\tilde{X}$ "with preprocessing". For this, the parameter J will typically be of the form $(I, G(I))$, where I is the original parameter and G is some polynomial-time map. Here $G(I)$ represents some auxiliary data computed from I.

For example, if we regard the Postnikov stage $P_k(Y)$ as parameterized by the finite simplicial complex Y, then by Definition 2.1, the algorithm for evaluating $\partial_i\sigma$ receives Y and σ as input. Thus, each time we want to know the ith face of some simplex, all of the computations are done from scratch.

In the algorithm from Theorem 1.2 for constructing a Postnikov system, we will proceed differently: given Y, we first compute, once and for all, some data based on Y, such as the first k homotopy groups of Y. Then we will represent P_k using these data (concretely, as a twisted product of suitable Eilenberg–MacLane spaces), instead of the "raw" representation by Y, so that these computations can be reused in all subsequent computations of face operators in P_k. This will make the computation of the face operators and other operations with the Postnikov system much more efficient, although if we care only about the distinction polynomial/non-polynomial, both ways are equivalent.

Keeping the parameters implicit. Although a locally polynomial-time simplicial set $(X(I) : I \in \mathcal{I})$ is defined as a mapping assigning a simplicial set $X(I)$ to every value of I, in most cases we can think of it as a single simplicial set X. The exact nature of the parameter I usually does not matter; it may be useful to keep in mind that X is actually parameterized, but in most of the subsequent discussion, we will suppress the parameter.

This is in agreement with the common practice in the literature on polynomial-time algorithms, where phrases like "the resulting graph has a polynomial size" are used, which are also formally imprecise but easily understood.

Converting finite simplicial complexes into simplicial sets. Here we make a slight digression and describe how a finite simplicial complex, which is one of the possible kinds of inputs for our algorithms, is (canonically) converted into a simplicial set.

Given a finite simplicial complex K, the corresponding simplicial set $\mathrm{SSet}(K)$, which in particular has the same geometric realization as K and thus specifies the same topological space, is defined as follows. The k-dimensional *nondegenerate* simplices of $\mathrm{SSet}(K)$ are just

the k-simplices of K, with the face operators defined in the obvious way. It remains to specify the degenerate simplices and the face and degeneracy operators on them. For this, we can use a standard fact about simplicial sets: every degenerate simplex τ can be expressed as $s_{i_t}s_{i_{t-1}}\cdots s_{i_1}\sigma$, where σ is a uniquely determined nondegenerate simplex of X and $i_1 < i_2 < \cdots < i_t$ is a uniquely determined sequence of integers. Thus, we can represent τ by σ and $i_1, \ldots, i_t$. With this representation, the face and degeneracy operators can be evaluated by simple rules; see, e.g., [12, 19]. (Also see [12, Section 3] for another, simpler way of adding degenerate simplices to a simplicial complex.)

Then $(\mathrm{SSet}(K) : K \in \mathcal{FSC})$ forms a locally polynomial-time simplicial set, whose parameter set $\mathcal{FSC}$ consists of all finite simplicial complexes.

More generally, we can consider the family of all finite simplicial sets, which are given by lists of nondegenerate simplices for each of the relevant dimensions and tables specifying the face operators, and where the degenerate simplices and degeneracy operators are represented as above. Then the identity map on $\mathcal{FSS}$ forms a locally polynomial-time simplicial set.

2.4 Reductions, strong equivalences, and polynomial-time homology

It turns out that the notion of locally polynomial-time simplicial set is too weak for most computational purposes. We can inspect such a simplicial set locally, but it is in general impossible to compute useful global information about it, such as homology groups or homotopy groups.

Here we introduce a stronger notion of simplicial set with polynomial-time homology, modeled after simplicial sets with effective homology due to Sergeraert et al. This is a (parameterized) locally polynomial-time simplicial set X whose normalized chain complex $C_*(X)$ is, moreover, associated with another, typically much smaller chain complex EC_*, which we can think of as a finitary approximation of $C_*(X)$. (The notation EC_* follows [35], and it should suggest that EC_* is an "effective version" of C_*.) The chain groups EC_k have polynomially many generators for every fixed k, and thus we can compute each homology group $H_k(EC_*)$ in polynomial time. The association of EC_* with $C_*(X)$ is such that these homology computations in EC_* can be "pulled back" to $C_*(X)$. We will now define the properties of EC_* and the way it is associated with $C_*(X)$ in detail.

Definition 2.3 (Globally polynomial-time chain complexes)**.** *A* globally polynomial-time chain complex *is a locally polynomial-time chain complex* $(EC(I)_* : I \in \mathcal{I})$ *such that, for each fixed k, the chain group $EC(I)_k$ is finitely generated, and there is an algorithm that, given $I \in \mathcal{I}$, outputs the list of elements of the distinguished basis $\mathrm{Bas}(I)_k$ of $EC(I)_k$, in time bounded by a polynomial in $\mathsf{size}(I)$ (and in particular, the rank of $EC(I)_k$ is bounded by a polynomial in $\mathsf{size}(I)$).*

We note that, for a globally polynomial-time EC_* and each fixed k, we can compute the matrix of the differential $d_k \colon EC_k \to EC_{k-1}$ w.r.t. the distinguished bases in polynomial time—we just evaluate d_k on each element of the distinguished basis Bas_k. Then the homology groups $H_k(EC_*)$ is computed using a Smith normal form algorithm applied to the matrices of d_k and d_{k+1}, as is explained in standard textbooks (such as [21]). Polynomial-time algorithms for the Smith normal form are nontrivial but known [16]; also see [44] for apparently the asymptotically fastest deterministic algorithm.

Globally polynomial-time Abelian groups. By the above, we can compute $H_k(EC_*)$

in polynomial time. We represent its isomorphism type[14] in the usual way, as a direct sum $\mathbb{Z}^r \oplus (\mathbb{Z}/m_1) \oplus (\mathbb{Z}/m_2) \oplus \cdots \oplus (\mathbb{Z}/m_s)$. But in our algorithms, we are not interested just in knowing this description of the homology group; we will also need to work with its elements, with homomorphisms into it, etc. Moreover, since the chain complex EC_* is parameterized, the homology group $H_k(EC_*)$ should be regarded as parameterized as well (and similarly for homotopy groups of parameterized simplicial sets). We thus define a globally polynomial-time Abelian group in analogy with a globally polynomial-time chain complex.

First, let $\mathcal{M}$ be the set of all $(r+s)$-tuples $\mathbf{m} = (0, 0, \ldots, 0, m_1, \ldots, m_s)$ specifying isomorphism types of finitely generated Abelian groups in the way introduced in Section 2.2. For $\mathbf{m} \in \mathcal{M}$, let $\mathrm{Ab}(\mathbf{m})$ be the group $\mathbb{Z}^r \oplus (\mathbb{Z}/m_1) \oplus \cdots \oplus (\mathbb{Z}/m_s)$, with elements represented by $(r+s)$-tuples $(\alpha_1, \ldots, \alpha_{r+s})$, $\alpha_1, \ldots, \alpha_r \in \mathbb{Z}$, $\alpha_{r+i} \in \mathbb{Z}/m_i$. Here $\mathrm{Ab}(\mathbf{m})$ can be regarded as a canonical representation of an Abelian group with the isomorphism type $\mathbf{m}$.

Now we define a *parameterized Abelian group* and *locally polynomial-time Abelian group* in an obvious analogy with the corresponding notions for simplicial sets and chain complexes. A *globally polynomial-time Abelian group* $(\pi(I) : I \in \mathcal{I})$ is a locally polynomial-time Abelian group equipped with a polynomial-time algorithm that, given $I \in \mathcal{I}$, returns an $\mathbf{m} \in \mathcal{M}$ specifying the isomorphism type of $\pi(I)$, and with a polynomial-time isomorphism of $\pi(I)$ with $\mathrm{Ab}(\mathbf{m})$. In more detail, in time polynomial in $\mathsf{size}(I)$ we can compute a basis $(b_1, b_2, \ldots, b_{r+s})$ of $\pi(I)$ such that b_i generates the ith cyclic summand isomorphic to $\mathbb{Z}$ (for $i \leq r$) or $\mathbb{Z}/m_{i-r}$ (for $i > r$) in an expression of $\pi(I)$ as a direct sum. Moreover, given an arbitrary element $a \in \pi(I)$, in time polynomial in $\mathsf{size}(I) + \mathsf{size}(a)$ we can compute the coefficients $\alpha_1, \ldots, \alpha_{r+s}$ such that $a = \sum_{i=1}^{r+s} \alpha_i b_i$. This provides the isomorphism $\pi(I) \to \mathrm{Ab}(\mathbf{m})$, and the inverse mapping is also obviously polynomial-time computable.

We now consider the globally polynomial-time chain complex EC_* parameterized by $\mathcal{I}$. We want to regard $H_k(EC_*)$ as a globally polynomial-time Abelian group parameterized by $\mathcal{I}$. To this end, we need that the computation of $H_k(EC(I)_*)$ returns its isomorphism type $\mathbf{m}$, and also fixes an isomorphism of $H_k(EC(I)_*)$ with $\mathrm{Ab}(\mathbf{m})$. Such an isomorphism is naturally obtained from the Smith normal form algorithm.[15] In this way, $H_k(EC_*)$ becomes a globally polynomial-time Abelian group parameterized by $\mathcal{I}$.

Moreover, given a chain $z \in Z_k(EC(I)_*)$, we can compute in polynomial time the corresponding homology class $[z] \in H_k(EC(I)_*)$. This defines a polynomial-time homomorphism $Z_k(EC_*) \to H_k(EC_*)$, also parameterized by $\mathcal{I}$. Slightly more generally, given a chain $c \in EC_k$, we can decide whether c is a cycle, and if yes, compute $[c]$. Moreover, if $[c]$ is zero, that is, if c is a boundary, we can also compute a "witness," i.e., a $(k+1)$-chain b with $c = d_{k+1}b$. Conversely, given $h \in H_k(EC_*)$, we can compute a *representing cycle*, i.e., $z \in Z_k(EC_*)$ with $[z] = h$. All of these calculations are easily done in polynomial time using the Smith normal form of the matrices of the differentials.

Reductions. Now we start discussing the way of associating a "small" chain complex EC_* with a "big" chain complex C_*. First we deal with the usual setting of homological algebra, where we consider individual chain complexes, rather than parameterized ones, and then we add some remarks on transferring the notions to the setting of parameterized chain complexes

[14] To get a bijective correspondence with isomorphism types, we should ask for divisibility $m_1 | \cdots | m_s$. We do not care about uniqueness, however, and thus we will not require this.

[15] Formally, for this we need the Smith normal form algorithm to be *deterministic*, so that it always returns the same isomorphism for a given I (which need not be true for a randomized algorithm, for example). However, in an actual implementation, this issue does not arise, since anyway we want to store in memory the Smith normal form once computed for a given I, in order to avoid repeated computations.

and maps.

The most common way in algebraic topology of making two chain complexes C_* and $\tilde{C}_*$ "equivalent" is *chain homotopy equivalence*, but for effective homology and polynomial-time homology, it is more convenient to use two special cases of chain homotopy equivalences, namely, *reduction*[16] and *strong equivalence*.

If $f, g\colon C_* \to \tilde{C}_*$ are two chain maps, then a *chain homotopy* of f and g is a sequence $(h_k)_{k\in\mathbb{Z}}$ of homomorphisms, where $h_k\colon C_k \to \tilde{C}_{k+1}$ (raising the dimension by one), such that $g_k - f_k = \tilde{d}_{k+1}h_k + h_{k-1}d_k$. Chain maps and chain homotopies can be regarded as algebraic counterparts of continuous maps of spaces and their homotopies, respectively. In particular, two chain-homotopic chain maps induce the same map in homology.

Definition 2.4 (Reduction)**.** *Let C_* and $\tilde{C}_*$ be chain complexes. A* reduction *ρ from C_* to $\tilde{C}_*$ consists of three maps $f = (f_k)_{k\in\mathbb{Z}}, g = (g_k)_{k\in\mathbb{Z}}, h = (h_k)_{k\in\mathbb{Z}}$, such that*

(i) *$f\colon C_* \to \tilde{C}_*$ and $g\colon \tilde{C}_* \to C_*$ are chain maps;*
(ii) *the composition $fg\colon \tilde{C}_* \to \tilde{C}_*$ is equal to the identity $\mathrm{id}_{\tilde{C}_*}$, while the composition $gf\colon C_* \to C_*$ is chain-homotopic to id_{C_*}, with $h\colon C_* \to C_*$ providing the chain homotopy, i.e. $\mathrm{id}_{C_*} - gf = dh + hd$; and*
(iii) *$fh = 0$, $hg = 0$, and $hh = 0$.*

We write

$$C_* \Rrightarrow \tilde{C}_*$$

if there is a reduction from C_* to $\tilde{C}_*$.

A reduction $C_* \Rrightarrow \tilde{C}_*$ can be depicted by the following diagram:

$$\begin{tikzcd} C_* \arrow[loop left, "h"] \arrow[r, bend left, "f"] & \tilde{C}_* \arrow[l, bend left, "g"] \end{tikzcd}$$

Intuitively, such a reduction is a tool that allows us to reduce questions about homology of a "big" chain complex C_* to questions about homology of a "smaller" chain complex $\tilde{C}_*$. In particular, the existence of a reduction $C_* \Rrightarrow \tilde{C}_*$ implies that $H_k(C_*) \cong H_k(\tilde{C}_*)$ for all k.

It is easily checked that (f, g, h) is a reduction $C_* \Rrightarrow \tilde{C}_*$ and (f', g', h') is a reduction $\tilde{C}_* \Rrightarrow \tilde{\tilde{C}}_*$, then there is a reduction $C_* \Rrightarrow \tilde{\tilde{C}}_*$, namely, $(f'f, gg', h + gh'f)$ (see, e.g., [35, Proposition 59]). We will also need a (straightforward) extension to composing a larger number of reductions (the proof is omitted).

Lemma 2.5. *Let $C_*^{(1)}, \ldots, C_*^{(n)}$ be chain complexes, and let $\rho^{(i)} = (f^{(i)}, g^{(i)}, h^{(i)})$ be a reduction $C_*^{(i)} \Rrightarrow C_*^{(i+1)}$, $i = 1, 2, \ldots, n-1$. Then the reduction $(f, g, h)\colon C_*^{(1)} \Rrightarrow C_*^{(n)}$ obtained by composing these reductions is given by $f = f^{(n-1)}f^{(n-2)}\cdots f^{(1)}$, $g = g^{(1)}g^{(2)}\cdots g^{(n-1)}$, and*

$$h = h^{(1)} + g^{(1)}h^{(2)}f^{(1)} + \cdots + g^{(1)}g^{(2)}\cdots g^{(n-2)}h^{(n-1)}f^{(n-2)}\cdots f^{(1)}.$$

Strong equivalences. While reductions $C_* \Rrightarrow \tilde{C}_* \Rrightarrow \tilde{\tilde{C}}_*$ compose to a reduction $C_* \Rrightarrow \tilde{\tilde{C}}_*$, in some constructions one naturally arrives at a different kind of situation:

$$C_* \Lleftarrow \tilde{\tilde{C}}_* \Rrightarrow \tilde{C}_*. \tag{2}$$

[16] In a part of the literature, e.g., in Eilenberg and Mac Lane [9, Section 12], the word *contraction* is used instead of reduction, while reduction has a different meaning.

Here we have no natural way of composing the reductions to obtain a reduction between C_* and $\tilde{C}_*$. For algorithmic purposes, we regard the situation (2) as a primitive notion, called *strong chain homotopy equivalence* or just strong equivalence.

Definition 2.6 (Strong equivalence)**.** *A* strong equivalence *of chain complexes C_* and $\tilde{C}_*$, in symbols $C_* \Lleftarrow\!\!\!\Rrightarrow \tilde{C}_*$, means that there exists another chain complex $\tilde{\tilde{C}}_*$ and reductions $C_* \Lleftarrow \tilde{\tilde{C}}_* \Rrightarrow \tilde{C}_*$.*

Lemma 2.7. *Strong equivalence is transitive: if $C_* \Lleftarrow\!\!\!\Rrightarrow \tilde{C}_*$ and $\tilde{C}_* \Lleftarrow\!\!\!\Rrightarrow \tilde{\tilde{C}}_*$, then $C_* \Lleftarrow\!\!\!\Rrightarrow \tilde{\tilde{C}}_*$.*

Proof. There are several proofs available. One of them follows [35, Proposition 124] (using the algebraic mapping cylinder). Another possibility is to regard reductions as special cases of chain homotopy equivalences, which are closed under composition, and then show that a chain homotopy equivalence can be converted into a strong equivalence, also using a suitable mapping cylinder—see, e.g., [3], [28, Sec. 3].

Here we offer yet another short proof. Let us consider strong equivalences $C_* \Lleftarrow A_* \Rrightarrow \tilde{C}_*$ and $\tilde{C}_* \Lleftarrow A'_* \Rrightarrow \tilde{\tilde{C}}_*$. In view of Lemma 2.5 it is suffices to exhibit a strong equivalence $A_* \Lleftarrow\!\!\!\Rrightarrow A'_*$.

Let the reduction $A_* \Rrightarrow \tilde{C}_*$ be (f, g, h) and let the reduction $A'_* \Rrightarrow \tilde{C}$ be (f', g', h'). We construct a new chain complex D_*, the *double mapping cylinder* of the pair of maps $A_* \xleftarrow{g} \tilde{C}_* \xrightarrow{g'} A'_*$ (this construction is analogous to the mapping cylinder introduced earlier). Its chain groups are

$$D_k := A_k \oplus \tilde{C}_{k-1} \oplus A'_k$$

and the differential is given by $d^D(a, c, a') := (d(a) - g(c), -\tilde{d}(c), d'(a') + g'(c))$ (where $d, \tilde{d}, d'$ are differentials in A_*, $\tilde{C}_*$, and A'_*, respectively). It is easily checked that D_* indeed forms a chain complex.

We now describe a reduction $(F, G, H)\colon D_* \Rrightarrow A_*$; we set

$$F(a, c, a') := a + gf'(a'),\ G(a) = (a, 0, 0),\ H(a, c, a') = (0, f'(a'), h'(a')).$$

The reduction $(F', G', H')\colon D_* \Rrightarrow A'_*$ is obtained almost symmetrically as

$$F'(a, c, a') := a' + g'f(a),\ G'(a') = (0, 0, a'),\ H'(a, c, a') = (h(a), -f(a), 0).$$

Checking that both (F, G, H) and (F', G', H') are indeed reductions is routine and we omit it. $\square$

Polynomial-time reductions and strong equivalences. Let $(C(I)_* : I \in \mathcal{I})$ and $(\tilde{C}(I)_* : I \in \mathcal{I})$ be two locally polynomial-time chain complexes with the *same* parameter set. A *polynomial-time reduction* of C_* to $\tilde{C}_*$, in symbols

$$C_* \overset{\mathrm{P}}{\Rrightarrow} \tilde{C}_*,$$

is a triple $\rho = (f, g, h)$. Here $f = (f_I)_{I \in \mathcal{I}}$ is a polynomial-time chain map $C_* \to \tilde{C}_*$, $g = (g_I)_{I \in \mathcal{I}}$ is a polynomial-time chain map $\tilde{C}_* \to C_*$, and $h = (h_I)_{I \in \mathcal{I}}$ is a polynomial-time chain homotopy $C_* \to C_*$, defined in obvious analogy with a polynomial-time chain map. For each I, (f_I, g_I, h_I) form a reduction $C(I)_* \Rrightarrow \tilde{C}(I)_*$ according to Definition 2.4.

Similarly, we define a polynomial-time strong equivalence of two locally polynomial-time chain complexes, $C_* \overset{\mathrm{P}}{\Longleftrightarrow} \tilde{C}_*$, with the middle chain complex also locally polynomial-time and with the same parameterization as C_* and $\tilde{C}_*$.

By the fact that a composition of any constant number of polynomial-time maps is polynomial-time, it is easy to check that the proof of Lemma 2.7 yields the following.

Corollary 2.8. *Polynomial-time strong equivalence of locally polynomial-time chain complexes is transitive:* $C_* \overset{\mathrm{P}}{\Longleftrightarrow} \tilde{C}_*$ *and* $\tilde{C}_* \overset{\mathrm{P}}{\Longleftrightarrow} \tilde{\tilde{C}}_*$ *implies* $C_* \Longleftrightarrow \tilde{\tilde{C}}_*$.

Polynomial-time homology. With the notions of polynomial-time strong equivalence and globally polynomial-time chain complex, the definition of polynomial-time homology is now straightforward.

Definition 2.9 (Chain complexes and simplicial sets with polynomial-time homology)**.** *We say that a parameterized chain complex C_* is equipped with* polynomial-time homology *if C_* is locally polynomial-time and there are a globally polynomial-time chain complex EC_* and a polynomial-time strong equivalence $C_* \overset{\mathrm{P}}{\Longleftrightarrow} EC_*$.*

A parameterized simplicial set X is equipped with polynomial-time homology if X is locally polynomial-time and its normalized chain complex $C_*(X)$ is equipped with polynomial-time homology.

We should perhaps stress that equipping a parameterized simplicial set X with polynomial-time homology does *not* mean only the ability of computing the homology groups of $X(I)$ in time polynomial in $\mathsf{size}(I)$ (for every fixed dimension); this ability is a consequence of polynomial-time homology, but in itself it would not be sufficient.

For one thing, if X is equipped with polynomial-time homology, we can do for $C_*(X)$ all of the computations mentioned after Definition 2.3: finding a representative of a given homology class, the homology class of a given chain, and a witness for being a boundary.

Moreover, the definition of polynomial-time homology, following the earlier notion of effective homology by Sergeraert et al., is designed so that it has the following meta-property: if $X^{(1)}, \ldots, X^{(t)}$ are simplicial sets equipped with polynomial-time homology and Φ is a "reasonable" way of constructing a new simplicial set from t old ones, then the simplicial set $\Phi(X^{(1)}, \ldots, X^{(t)})$ can also be equipped with polynomial-time homology (some of the constructions also involve polynomial-time simplicial maps, polynomial-time chain maps, etc.). Of course, this is only a guiding principle, and for every specific construction Φ used in our algorithm, we need a corresponding result about preserving polynomial-time homology by Φ. The next section is devoted to such results.

The reader may also wonder what are homology computations good for in algorithms for computing homotopy groups and Postnikov systems. The connection is via the *Hurewicz isomorphism*, which in its simplest form asserts that, for a 1-connected space Y, the first nonzero homotopy group of Y occurs in the same dimension as the first nonzero homology group, and these two groups are isomorphic. Thus, roughly speaking, to find $\pi_k(Y)$, the Postnikov system algorithm "kills" the first $k-1$ homotopy groups of Y by constructing the mapping cone of $\varphi_{k-1}\colon Y \to P_{k-1}$ with polynomial-time homology, and then it computes the appropriate homology group of this cone.

Let us remark that in [18], polynomial-time homology was defined using only reductions, rather than strong equivalences (since strong equivalences were not needed there). Of course, a reduction is a special case of strong equivalence, so the definition here is more permissive.

3 A toolbox of operations for polynomial-time homology

In this longish section we will build a repertoire of algorithmic operations on simplicial sets and chain complexes, in such a way that if the input objects come with polynomial-time homology, the output object is also equipped with polynomial-time homology.

As was mentioned in the introduction, we mostly review known methods, developed for effective homology and based on much older work by algebraic topologists. We try to make the presentation streamlined and mostly self-contained, and in particular, we describe the algorithms in full, sometimes referring to the literature for details of proofs. Moreover, there are places where polynomiality requires extra analysis or assumptions; most notably, Section 3.1 (products of many factors) and Section 3.8 (polynomial-time homology for $K(\mathbb{Z}/m, 1)$) contain some new material.

For the rest of the paper, we will use only three specific results of this section: Proposition 3.8 (mapping cone), Corollary 3.18 (a certain pullback operation), and Theorem 3.16 (polynomial-time homology for Eilenberg–MacLane spaces). But we will also need some of the notions and simple facts introduced here.

Let us remark that some of the operations can be implemented in several different ways. For example, polynomial-time homology for $K(\mathbb{Z}/m, 1)$ can most likely be obtained directly by modifying the method of [18] used for $K(\mathbb{Z}, 1)$, and for the passage from $K(\pi, k)$ to $K(\pi, k+1)$, one could also use the method in [25, Chap. 4] (also see [1]). Our main criterion for selecting among the various possibilities was simplicity of presentation and general applicability of the tools.

3.1 Products

We recall that the *product* $X \times Y$ of simplicial sets X and Y is the simplicial set whose k-simplices are ordered pairs (σ, τ), where $\sigma \in X_k$ and $\tau \in Y_k$. The face and degeneracy operators are applied to such pairs componentwise. We have $|X \times Y| \cong |X| \times |Y|$ for geometric realizations.[17] The definition of the product is deceptively simple, but actually it hides a sophisticated way of triangulating the product (and degenerate simplices play a crucial role)—see [35] or [12] for an explanation.

As shown by Sergeraert et al. as one of the first steps in the theory of effective homology, if effective homology is available for X and Y, then it can also be obtained for $X \times Y$. The core of this result is the Eilenberg–Zilber theorem (see, e.g., [35, Theorem 123]), which provides a reduction of $C_*(X \times Y)$ to the tensor product $C_*(X) \otimes C_*(Y)$, and which goes back to Eilenberg and Mac Lane [9, 10]. The proof immediately shows that polynomial-time homology for X, Y yields polynomial-time homology for $X \times Y$.

However, this works directly only for products of two, or constantly many, factors, while we need to deal with products $X^{(1)} \times \cdots \times X^{(n)}$ of arbitrarily many factors. There the situation with polynomiality is somewhat more subtle, and we will actually need an additional condition on the $X^{(i)}$'s in order to obtain polynomial-time homology. We begin with defining the notion needed for the extra condition.

Definition 3.1 (k-reduced)**.** *A simplicial set X is* k-reduced*, where $k \geq 0$ is an integer, if X has a single* 0*-simplex (vertex) and no nondegenerate simplices of dimensions* 1 *through k. We call a chain complex C_** k-reduced *if $C_0 \cong \mathbb{Z}$ and $C_i = 0$ for $1 \leq i \leq k$.*

[17] To be precise, the product of topological spaces on the right-hand side should be taken in the category of k-spaces; but for the spaces we encounter, it is the same as the usual product of topological spaces.

We remark that k-reducedness is a very useful property of simplicial sets, which has no analog for simplicial complexes. For example, being k-reduced is an easily checkable certificate for k-connectedness.

Proposition 3.2 (Product with many factors)**.** *Let $(X(I) : I \in \mathcal{I})$ be a simplicial set with polynomial-time homology. Let us form a new parameter set $\mathcal{J} = \bigcup_{n=1}^{\infty} \mathcal{I}^n$, where $\mathcal{I}^n$ is the n-fold Cartesian product, and let $(W(J) : J \in \mathcal{J})$ be the parameterized simplicial set of products, with $W(I_1, I_2, \ldots, I_n) := X(I_1) \times \cdots \times X(I_n)$. For $J = (I_1, \ldots, I_n) \in \mathcal{J}$, let $\mathsf{size}(J) = \sum_{i=1}^{n} \mathsf{size}(I_i)$, and for a simplex $\sigma = (\sigma_1, \ldots, \sigma_n) \in W(J)$, let $\mathsf{size}(\sigma) = \sum_{i=1}^{n} \mathsf{size}(\sigma_i)$. Let us also assume that all the $X(I)$ and all the chain complexes witnessing polynomial-time homology for X are 0-reduced. Then W can be equipped with polynomial-time homology.*

For reasons of "uniform polynomiality", we needed to assume that the factors in the considered products are all instances of a single parameterized simplicial set. However, as we remarked above, the product of a *constant* number of arbitrary, possibly different, simplicial sets with polynomial-time homology can be equipped with polynomial-time homology. This allows us to obtain polynomial-time homology for products where all but a constant number of factors are 0-reduced and come from the same parameterized simplicial set, while the remaining factors are arbitrary.

In the forthcoming proof, for brevity, we are going to write $X^{(i)}$ instead of $X(I_i)$, and use similar abbreviations for chain complexes.

Tensor products. Before discussing the proof, we need some preparations concerning tensor products. Let $C_*^{(1)}$ and $C_*^{(2)}$ be chain complexes, and suppose, as we do for locally polynomial-time chain complexes, that each chain group $C_k^{(i)}$ has a distinguished basis $\mathrm{Bas}_k^{(i)}$. Then the *tensor product* $T_* := C_*^{(1)} \otimes C_*^{(2)}$ can be defined as the chain complex in which T_k is the free Abelian group over the distinguished basis

$$\mathrm{Bas}_k := \{b_1 \otimes b_2 : b_1 \in \mathrm{Bas}_{k_1}^{(1)}, b_2 \in \mathrm{Bas}_{k_1}^{(2)}, k_1 + k_2 = k\}.$$

Here we may regard $b_1 \otimes b_2$ just as a formal symbol. For arbitrary chains $c_1 \in C_{k_1}^{(1)}$, $c_2 \in C_{k_2}^{(2)}$, $k_1 + k_2 = k$, the k-chain $c_1 \otimes c_2$ is then defined using linearity of $\otimes$ in both operands, as the appropriate linear combination of the elements of Bas_k.

The differential in T_* is given on the elements of Bas_k by

$$d_k(b_1 \otimes b_2) := d_{k_1}^{(1)}(b_1) \otimes b_2 + (-1)^{k_1} b_1 \otimes d_{k_2}^{(2)}(b_2), \tag{3}$$

where as above, $k_i = \deg(b_i)$.

Next, let us consider the tensor product $T_* := C_*^{(1)} \otimes \cdots \otimes C_*^{(n)}$ of many factors. The distinguished basis Bas_k now consists of elements $b_1 \otimes \cdots \otimes b_n$, with each b_i an element of a distinguished basis in $C_*^{(i)}$, $\sum_{i=1}^{n} \deg(b_i) = k$. Hence the rank of T_k equals

$$\mathrm{rank}(T_k) = \sum_{k_1 + \cdots + k_n = k} \prod_{i=1}^{n} \mathrm{rank}(C_{k_i}^{(i)}). \tag{4}$$

Thus, if many of the $C^{(i)}$ are not 0-reduced, already $\mathrm{rank}(T_0)$ is exponentially large; for example, if each $C_0^{(i)}$ is $\mathbb{Z} \oplus \mathbb{Z}$, then $\mathrm{rank}(T_0) = 2^n$. This is the basic reason why we need the

0-reducedness conditions in Proposition 3.2. If, on the other hand, all the $C_*^{(i)}$'s are 0-reduced, then so is T_*.

The key to the polynomial-time bounds we need is the following lemma.

Lemma 3.3. *Let $(C(I)_* : I \in \mathcal{I})$ be a locally polynomial-time chain complex, with all the $C(I)_*$ 0-reduced, let $\mathcal{J}$ be the parameter set as in Proposition 3.2, and let $(T(J)_* : J \in \mathcal{J})$ be the parameterized set of tensor products, with $T(I_1, \ldots, I_n)_* = C_*^{(1)} \otimes \cdots \otimes C_*^{(n)}$ (where $C_*^{(i)}$ abbreviates $C(I_i)_*$), and with the same definitions of encoding sizes as in Proposition 3.2. Then T_* is also 0-reduced and locally polynomial-time, and given chains $c_i \in C_{k_i}^{(i)}$ with $\sum_{i=1}^{n} k_i = k$, the k-chain $c_1 \otimes \cdots \otimes c_n$ can be computed (i.e., expressed in the distinguished basis of $T_k(J)$) in time polynomial in $\mathsf{size}(J) + \sum_{i=1}^{n} \mathsf{size}(c_i)$, assuming k fixed.*

Proof. To show that the differential d_k of T_* is a polynomial-time map, it is enough to consider computing it on elements $b_1 \otimes \cdots \otimes b_n$ of the standard basis. By iterating the differential formula (3), we can express $d_k(b_1 \otimes \cdots \otimes b_n)$ as a sum of n terms of the form $\pm c_1 \otimes \cdots \otimes c_n$, where each c_i is either b_i or $d_{k_i}(b_i)$. For evaluating this sum it is thus sufficient to be able to evaluate $c_1 \otimes \cdots \otimes c_n$ in polynomial time, as in the second claim of the lemma.

As for this second claim, we use the observation that if $\deg(c_1 \otimes \cdots \otimes c_n) = k$, then all but at most k of the c_i's have degree 0. Suppose that only $c_1, \ldots, c_k$ have nonzero degrees. Then we can compute $c_1 \otimes \cdots \otimes c_k$ in a straightforward way (at most $\prod_{i=1}^{k} \mathsf{size}(c_i)$ basis elements are involved, which is polynomially bounded for fixed k). Then the tensor product of the result with $c_{k+1} \otimes \cdots \otimes c_n$ amounts just to multiplying all coefficients by a number (since $C_0^{(k+1)} \cong \cdots \cong C_0^{(n)} \cong \mathbb{Z}$ by the 0-reducedness assumption) and renaming the basis elements appropriately. □

Proof of Proposition 3.2. We basically follow a proof for the case of effective homology (where it is enough to deal with two factors). There are two main steps, encapsulated in the following two lemmas, which together imply the proposition via Corollary 2.8 (composing strong equivalences).

Lemma 3.4 (Tensor product of strong equivalences)**.** *Let $(C(I)_* : I \in \mathcal{I})$ and $(\hat{C}(I)_* : I \in \mathcal{I})$ be a locally polynomial-time chain complexes, let $(EC(I)_* : I \in \mathcal{I})$ be a globally polynomial-time chain complex, and suppose that a strong equivalence $C_* \overset{\mathrm{P}}{\Lleftarrow} \hat{C}_* \overset{\mathrm{P}}{\Rrightarrow} EC_*$ is given, with all the chain complexes involved 0-reduced. As in Lemma 3.3, let T_*, $\hat{T}_*$, ET_* be the parameterized chain complexes of tensor products with factors from C_*, $\hat{C}_*$, and EC_*, respectively. Then ET_* is globally polynomial-time and there is a strong equivalence $T_* \overset{\mathrm{P}}{\Lleftarrow} \hat{T}_* \overset{\mathrm{P}}{\Rrightarrow} ET_*$.*

Lemma 3.5 (Eilenberg–Zilber for many factors)**.** *Let $(X(I) : I \in \mathcal{I})$ be a 0-reduced locally polynomial-time simplicial set, let $(W(J) : J \in \mathcal{J})$ be the parameterized set of products as in Proposition 3.2, and let $(T(J)_* : J \in \mathcal{J})$ be the parameterized chain complex of the tensor products $C_*(X^{(1)}) \otimes \cdots \otimes C_*(X^{(n)})$ as in Lemma 3.3. Then there is a polynomial-time reduction $C_*(W) \overset{\mathrm{P}}{\Rrightarrow} T_*$.*

Proof of Lemma 3.4. We know from Lemma 3.3 that T_*, $\hat{T}_*$, and ET_* are locally polynomial-time. To check that ET_* is globally polynomial-time, let us consider the chain group $ET(J)_k$, $J = (I_1, \ldots, I_n)$. Since EC_* is globally polynomial-time, there is a polynomial p such that $\mathrm{rank}(EC(I_i)_j) \le p(\mathsf{size}(I_i)) \le p(\mathsf{size}(J))$ for all J and all $j \le k$. Setting $N := p(\mathsf{size}(J))$, by

the 0-reducedness assumption and the rank formula (4) we get $\operatorname{rank}(ET(J)_k) \le \binom{n+k-1}{k} N^k$, which is bounded by a polynomial in $\mathsf{size}(J) \ge n$. Generating the distinguished basis of $ET(J)_k$ in polynomial time is done by a straightforward combinatorial enumeration algorithm. We conclude that ET_* is globally polynomial-time.

It remains to provide a polynomial-time reduction $\hat{T}_* \overset{\mathrm{P}}{\Rrightarrow} T_*$ (then $\hat{T}_* \overset{\mathrm{P}}{\Rrightarrow} ET_*$ is obtained in the same way). We consider $\hat{T}_*(J) = \hat{C}^{(1)} \otimes \cdots \otimes \hat{C}^{(n)}$, $J = (I_1, \ldots, I_n)$, $\hat{C}^{(i)}_* = \hat{C}_*(I_i)$, and let $\rho^{(i)} = (F^{(i)}, G^{(i)}, H^{(i)})$ be the reduction $\hat{C}^{(i)}_* \Rrightarrow C^{(i)}_*$ obtained from the assumption $\hat{C}_* \overset{\mathrm{P}}{\Rrightarrow} C_*$ (we use capital letters to avoid conflict with the notation of Lemma 2.5). The desired reduction $\hat{T}_*(J) \Rrightarrow T_*(J)$ goes through the intermediate chain complexes

$$\hat{C}^{(1)}_* \otimes \cdots \otimes \hat{C}^{(i-1)} \otimes C^{(i)}_* \otimes \cdots \otimes C^{(n)}_*, \quad i = 1, \ldots, n,$$

and the ith of these chain complexes is reduced to the $(i+1)$st one with the reduction that is the tensor product with ρ_i as the ith factor and the identities in all the other factors.

Specializing the formulas from Lemma 2.5 for composing reductions, we obtain the reduction $(F_J, G_J, H_J) : \hat{C}_*(J) \Rrightarrow C_*(J)$ with $F_J = F^{(1)} \otimes \cdots \otimes F^{(n)}$, $G_J = G^{(1)} \otimes \cdots \otimes G^{(n)}$, and

$$\begin{aligned} H_J \quad = \quad & H^{(1)} \otimes \mathrm{id} \otimes \cdots \otimes \mathrm{id} + G^{(1)} F^{(1)} \otimes H^{(2)} \otimes \mathrm{id} \otimes \cdots \otimes \mathrm{id} + \cdots \\ & + G^{(1)} F^{(1)} \otimes \cdots \otimes G^{(n-1)} F^{(n-1)} \otimes H^{(n)}. \end{aligned}$$

(Tensor products of chain maps are defined as expected, via $(f \otimes g)(a \otimes b) = f(a) \otimes g(b)$; for chain homotopies there is a sign convention involved, with the signs obviously polynomial-time computable—see, e.g., [35, Definition 57].)

These formulas define the desired reduction $(F_J, G_J, H_J)_{J \in \mathcal{J}} : \hat{T}_* \overset{\mathrm{P}}{\Rrightarrow} T_*$; polynomial-time computability of these maps follows from Lemma 3.3. □

Proof of Lemma 3.5. For the binary case, with simplicial sets Y and Z, there is the classical Eilenberg–Zilber reduction $C_*(Y \times Z) \Rrightarrow C_*(Y) \otimes C_*(Z)$, which is denoted by (AW, EML, SHI) (these are acronyms for Alexander–Whitney, Eilenberg–MacLane, and Shih[18]). Explicit formulas for these maps are available; see [14, pp. 1212–1213] (for AW and EML we also provide the formulas below). In particular, it is clear from these formulas that the maps AW, EML, SHI are polynomial-time for locally polynomial-time Y and Z.

To build the reduction $C_*(W(J)) \Rrightarrow T_*(J)$, where as usual $J = (I_1, \ldots, I_n)$, $W(J) = X^{(1)} \times \cdots \times X^{(n)}$, and $T_*(J) = C_*(X^{(1)}) \otimes \cdots \otimes C_*(X^{(n)})$, we go through the intermediate chain complexes

$$D^{(i)}_* := C_*(X^{(1)}) \otimes \cdots \otimes C_*(X^{(i-1)}) \otimes C_*(X^{(i)} \times \cdots \times X^{(n)}).$$

Let $(f^{(i)}, g^{(i)}, h^{(i)})$ be the reduction $D^{(i)}_* \Rrightarrow D^{(i+1)}_*$. We have $f^{(i)} = \mathrm{id} \otimes \cdots \otimes \mathrm{id} \otimes \mathrm{AW}^{(i)}$, $g^{(i)} = \mathrm{id} \otimes \cdots \otimes \mathrm{id} \otimes \mathrm{EML}^{(i)}$, and $h^{(i)} = \mathrm{id} \otimes \cdots \otimes \mathrm{id} \otimes \mathrm{SHI}^{(i)}$, where $(\mathrm{AW}^{(i)}, \mathrm{EML}^{(i)}, \mathrm{SHI}^{(i)})$ is the Eilenberg–Zilber reduction $C_*(X^{(i)} \times Z^{(i)}) \Rrightarrow C_*(X^{(i)}) \otimes C_*(Z^{(i)})$, with $Z^{(i)} := X^{(i+1)} \times \cdots \times X^{(n)}$.

Now $f^{(i)}, g^{(i)}, h^{(i)}$ are polynomial-time by Lemma 3.3, and so in order to verify the polynomiality of the composed reduction, using the formula in Lemma 2.5, it suffices to check

[18] The explicit formula for the operator SHI was found by Rubio [32] and proved by Morace—see the appendix in [27].

polynomiality of the compositions $f^{(i)}f^{(i-1)}\cdots f^{(1)}$ and $g^{(1)}g^{(2)}\cdots g^{(i)}$, $i = 1, 2, \ldots, n-1$. For simpler notation, we will discuss only the case $i = n-1$, but the case of arbitrary i is the same.

Let (σ_i, τ_i) be a k-dimensional simplex of $X^{(i)} \times Z^{(i)}$, which we also consider as a generator of $C_*(X^{(i)} \times Z^{(i)})$. According to [14], we have

$$\mathrm{AW}^{(i)}(\sigma_i, \tau_i) = \sum_{j=0}^{k} \partial_{j+1}\cdots\partial_k\sigma_i \otimes \partial_0\cdots\partial_{j-1}\tau_i.$$

Composing $f^{(2)}$ and $f^{(1)}$ thus yields

$$\begin{aligned} f^{(2)}f^{(1)}(\sigma_1, \sigma_2, \tau_3) \quad = \quad \sum_{0\le j_1+j_2\le k} \Big(&\partial_{j_1+1}\cdots\partial_k\sigma_1 \otimes \partial_0\cdots\partial_{j_1-1}\partial_{j_2+1}\cdots\partial_{k-j_1}\sigma_2 \\ &\otimes \partial_0\cdots\partial_{j_1-1}\partial_0\cdots\partial_{j_2-1}\tau_3\Big). \end{aligned}$$

Continuing in a similar manner, we obtain $f^{(n-1)}\cdots f^{(1)}(\sigma_1, \ldots, \sigma_n)$ as the sum

$$\sum_{0\le j_1+\cdots+j_{n-1}\le k} \sigma_1' \otimes \cdots \otimes \sigma_n',$$

where each σ_i' is the result of applying some number (at most k) of face operators to σ_i. The number of terms in this sum is $\binom{n+k-1}{k}$, which is polynomially bounded for k fixed, and each term is polynomial-time computable. Thus, the compositions $f^{(i)}\cdots f^{(1)}$ are polynomial-time computable.

Concerning the $g^{(i)}$'s, for the mapping $\mathrm{EML}^{(i)}$ we have, again following [14], for a p-simplex σ and a q-simplex τ, $p + q = k$,

$$\mathrm{EML}^{(i)}(\sigma \otimes \tau) = \sum_{\substack{\alpha,\beta:\alpha\cup\beta=\{0,1,\ldots,k-1\}\\ |\alpha|=q,|\beta|=p,\alpha\cap\beta=\emptyset}} \pm(s_\alpha\sigma, s_\beta\tau),$$

where, writing $\alpha = \{j_1, j_2, \ldots, j_q\}$, $j_1 < j_2 < \cdots < j_q$, s_α denotes the composition $s_{j_q}s_{j_{q-1}}\cdots s_{j_1}$ of degeneracy operators, and similarly for s_β. The sign $\pm$ depends on α and β in a simple way, and we do not want to bother the reader with specifying it (see [14]).

By iterating this formula, we find that, for a k-simplex $\sigma_1 \otimes \cdots \otimes \sigma_n$, where $\dim\sigma_i = k_i$, $k_1 + \cdots + k_n = k$,

$$g^{(1)}g^{(2)}\cdots g^{(n-1)}(\sigma_1 \otimes \cdots \otimes \sigma_n) = \sum_{\alpha_1,\ldots,\alpha_n} \pm(s_{\alpha_1}\sigma_1, s_{\alpha_2}\sigma_2, \ldots, s_{\alpha_n}\sigma_n),$$

where the sum is over certain choices of index sets $\alpha_1, \ldots, \alpha_n \subseteq \{0, 1, \ldots, k-1\}$. We need not specify these choices precisely here (it suffices to know that there is a polynomial-time algorithm for generating them); we just note that $|\alpha_i| = k - k_i$, since each of the simplices $s_{\alpha_i}\sigma_i$ must have dimension k. Therefore, the number of terms in the sum is bounded above by

$$\prod_{i=1}^{n} \binom{k}{k-k_i} = \prod_{i=1}^{n} \binom{k}{k_i} < 2^{k^2},$$

since there are at most k nonzero k_i's, and $\binom{k}{k_i} < 2^k$ always. (A more refined estimate gives a better bound, but still exponential in k.) So the number of terms depends only on k, and thus it is a constant in our setting.

This concludes the proof. □

3.2 Perturbation lemmas

The following situation often occurs in the theory of effective homology. Suppose that we have already managed to obtain a reduction $C_* \Rrightarrow \tilde{C}_*$ for some chain complexes C_* and $\tilde{C}_*$. Now we want a reduction from C'_* to some $\tilde{C}'_*$, where C'_* is a chain complex that is "similar" to C_*, in the following way: the chain groups of C_* and of C'_* are the same, i.e., $C_k = C'_k$ for all k, and the differential d' of C'_* is of the form $d' = d + \delta$, where d is the differential in C_*, and δ is a map that is "small" in a sense to be specified in Theorem 3.6 below. Thus, we regard d' as a *perturbation* of d.

In this setting, we would like to modify the differential $\tilde{d}$ in $\tilde{C}_*$ to a suitable $\tilde{d}'$, obtaining a new chain complex $\tilde{C}'_*$ and a reduction $C'_* \Rrightarrow \tilde{C}'_*$. If, for example, $\tilde{C}_*$ was globally polynomial-time, and the original reduction $C_* \Rrightarrow \tilde{C}_*$ provided polynomial-time homology for C_*, we would like the new reduction $C'_* \Rrightarrow \tilde{C}'_*$ to give polynomial-time homology for C'_*.

A tool for that is the *basic perturbation lemma*, originally discovered by Shih.[19] For our purposes, we formulate a version of the basic perturbation lemma which yields polynomial-time reductions.

To state it, we need a definition. Let $f\colon C_* \to C_*$ be a chain map of a chain complex into itself. We say that f is *nilpotent* if for every $c \in C_k$, $k \in \mathbb{Z}$, there is some n such that $(f_k)^n(c) = 0$, where $(f_k)^n$ is the n-fold composition of f_k with itself. Now if C_* is a parameterized chain complex, we say that f has *constant nilpotency bounds* if for every k there exists $N = N_k$, depending on k but not on the value of the parameter, such that $(f_k)^N$ is the zero map.

Theorem 3.6 (Basic perturbation lemma)**.** *Let (f, g, h) be a reduction $C_* \Rrightarrow \tilde{C}_*$, let C'_* be a chain complex with $C'_k = C_k$ for all k and with differential d', and let us set $\delta := d' - d$. If the composed map $h\delta$ is nilpotent, then there is a chain complex $\tilde{C}'_*$ with the same chain groups as $\tilde{C}_*$ and with a modified differential $\tilde{d}'$, and a reduction $C'_* \Rrightarrow \tilde{C}'_*$.*

If C_* and $\tilde{C}_*$ are locally polynomial-time chain complexes, (f, g, h) is a polynomial-time reduction, δ is a polynomial-time map, and the composition $h\delta$ has constant nilpotency bounds, then $\tilde{d}'$ is polynomial-time and $C'_* \overset{\mathrm{P}}{\Rrightarrow} \tilde{C}'_*$.

Proof. The proof of the existence statement, presented, e.g., in [35, Theorem 50], provides explicit formulas for $\tilde{d}'$ and for the desired reduction $(f', g', h') : C'_* \Rrightarrow \tilde{C}'_*$. Namely, using auxiliary chain maps φ and ψ defined by

$$\varphi := \sum_{i=0}^{\infty} (-1)^i (h\delta)^i, \qquad \psi := \sum_{i=0}^{\infty} (-1)^i (\delta h)^i,$$

we have $\tilde{d}' := \tilde{d} + f\psi\delta g$, $f' := f\psi$, $g' = \varphi g$, and $h' := \varphi h$. If $h\delta$ has constant nilpotency bounds, then so has δh, and for each fixed k, the number of nonzero term in the sums defining $\varphi(c)$ and $\psi(c)$, with $c \in C_k$, is bounded by a constant depending only on k but not on c. The claim about polynomiality follows. □

The basic perturbation lemma propagates the perturbation of the differential in the direction of the reduction arrow. If we have a strong equivalence $C_* \Lleftarrow \tilde{\tilde{C}}_* \Rrightarrow \tilde{C}_*$ and we want to perturb the differential of C_*, we first need to propagate the perturbation to $\tilde{\tilde{C}}_*$, i.e.,

[19] Let us remark that there are many variants, extensions, and generalizations of the basic perturbation lemma in the literature, whose usefulness is by far not restricted to an algorithmic context.

against the direction of the reduction. The next lemma tells us that this can always be done; actually, only the differential in $\tilde{\tilde{C}}_*$ needs to be modified, the reduction stays the same. We omit the easy proof—see [35, Proposition 49].

Lemma 3.7 (Easy perturbation lemma)**.** *Let $(f, g, h) : C_* \Rrightarrow \tilde{C}_*$ be a reduction, and let $\tilde{C}'_*$ be obtained from $\tilde{C}_*$ by perturbing the original differential $\tilde{d}$ to $\tilde{d}' = \tilde{d} + \tilde{\delta}$. Then (f, g, h) is a reduction $C'_* \Rrightarrow \tilde{C}'_*$, where C'_* is obtained from C_* by perturbing the original differential d to $d' := d + g\tilde{\delta}f$.*

Thus, under favorable circumstances, if a parameterized chain complex C_* is equipped with polynomial-time homology, the combination of the easy perturbation lemma and the basic one allows us to obtain polynomial-time homology for the perturbed chain complex C'_*.

3.3 Mapping cone

Here we consider the mapping cone operation for chain complexes, as introduced in Section 2.1.

Proposition 3.8 (Algebraic mapping cone)**.** *If $C_*, \tilde{C}_*$ are (parameterized) chain complexes with polynomial-time homology and $\varphi\colon C_* \to \tilde{C}_*$ is a polynomial-time chain map, then the cone* $\mathrm{Cone}_*(\varphi)$ *can be equipped with polynomial-time homology.*

Proof (sketch). This is essentially [35, Theorem 79]. We sketch the proof since it is a simple and instructive use of the perturbation lemmas.

Given strong equivalences $C_* \overset{\mathrm{P}}{\Lleftarrow\!\Rrightarrow} EC_*$ and $\tilde{C}_* \overset{\mathrm{P}}{\Lleftarrow\!\Rrightarrow} \widetilde{EC}_*$, we want to construct a polynomial-time strong equivalence of $\mathrm{Cone}_*(\varphi)$ with a suitable globally polynomial-time chain complex EM_*.

We observe that, by definition, the chain groups of $\mathrm{Cone}_*(\varphi)$ depend only on $C_*, \tilde{C}_*$ but not on φ (only the differential depends on φ). We thus first consider $\mathrm{Cone}_*(0_{C_* \to \tilde{C}_*})$, where $0_{C_* \to \tilde{C}_*}$ is the zero chain map of the indicated chain complexes. Given the strong equivalences for C_* and $\tilde{C}_*$ as above, it is straightforward to construct a strong equivalence

$$\mathrm{Cone}_*(0_{C_* \to \tilde{C}_*}) \overset{\mathrm{P}}{\Lleftarrow\!\Rrightarrow} \mathrm{Cone}_*(0_{EC_* \to \widetilde{EC}_*});$$

this is just a direct sum construction.

Next, we regard $\mathrm{Cone}_*(\varphi)$ as a perturbation of $\mathrm{Cone}_*(0_{C_* \to \tilde{C}_*})$. Then we propagate the perturbation through the strong equivalence; in the application of the basic perturbation lemma, it turns out that the nilpotency of the relevant maps is bounded by 2 (independent of k). We refer to [35, Theorems 61,79] for details. □

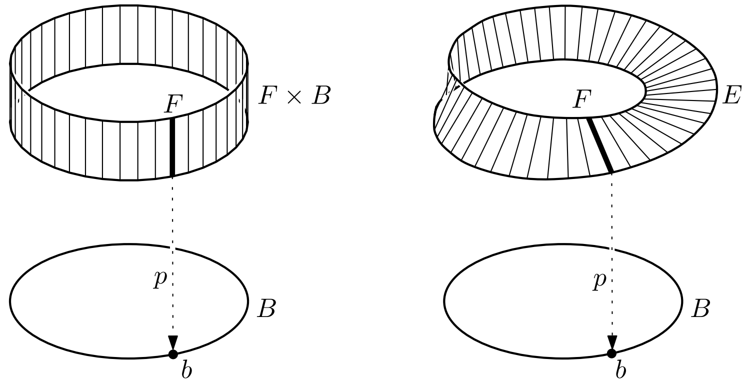
We remark that the strong equivalence $\mathrm{Cone}_*(\varphi) \overset{\mathrm{P}}{\Lleftarrow\!\Rrightarrow} EM_*$ produced in the proposition restricts to the original strong equivalence $\tilde{C}_* \overset{\mathrm{P}}{\Lleftarrow\!\Rrightarrow} \widetilde{EC}_*$. This follows at once from the explicit formulas in the perturbation lemmas and the fact that the involved perturbation is zero on $\tilde{C}_*$.

3.4 Twisted product

On fiber bundles. Our main goal is the computation of a Postnikov system for a given space Y. As we have mentioned, the kth stage of a Postnikov system can be thought of as

an approximation of Y, in a homotopy-theoretic sense, made of simple building blocks, which are called Eilenberg–MacLane spaces. These building blocks will be discussed in Section 3.7 below, but here we will consider the operation used to paste the building blocks together.

To convey some intuition, we begin with the topological notion of *fiber bundle*[20] (a *vector bundle* is a special case of a fiber bundle). Let B, the *base space*, and F, the *fiber space*, be two spaces. The Cartesian product $F \times B$ can be thought of as a copy of F sitting above each point of B; for B the unit circle S^1 and F a segment this is indicated in the left picture:



The product $F \times B$ is a *trivial* fiber bundle, while the right picture shows a nontrivial fiber bundle (a Möbius band in this case). Above every point $b \in B$, we still have a copy of F, and moreover, each such b has a small neighborhood U such that the union of all fibers sitting above U is homeomorphic to the product $F \times U$, a rectangle in the picture. However, globally, the union of the fibers above all of B forms a space E, the *total space* of the fiber bundle, that is in general different from $F \times B$.

More precisely, a fiber bundle is given as $p\colon E \to B$, where E, B are spaces and p is a surjective map, such that for every $b \in B$ there are a neighborhood U of b and a homeomorphism $h\colon p^{-1}(U) \to F \times U$ fixing the second component, i.e., with $h(x)_2 = p(x)$ for every $x \in E$. (Other famous examples of nontrivial fiber bundles involve the the Klein bottle with $B = F = S^1$ or the Hopf fibration $S^3 \to S^2$.)

For our purposes, we will deal with fiber bundles where the fiber F has "enough symmetries," meaning that there is a group G acting on the fiber F, and this helps in specifying the total space E in terms of B, F, and some additional data which, informally speaking, tell us how E is "twisted" compared to the product $F \times B$.

Simplicial groups. In order to define the appropriate simplicial notions, we first need to recall that a *simplicial group* is a simplicial set G such that, for each $k \geq 0$, the set G_k of k-dimensional simplices forms a group, and moreover, the face and degeneracy operators are group homomorphisms.

A *parameterized simplicial group* and a *locally polynomial-time simplicial group* are defined in an obvious analogy with the corresponding notions for simplicial sets and chain complexes.

A basic example of a simplicial group is the standard simplicial model of an Eilenberg–MacLane space; see Section 3.7 below. Actually, it is known that every Abelian simplicial group is homotopy equivalent to a product of Eilenberg–MacLane spaces (see [19, Chap. V]),

[20] In the literature on simplicial sets, effective homology and such, one usually speaks about a *fibration*, which is a notion more general than a fiber bundle; roughly speaking, a fibration can be regarded as a "fiber bundle up to homotopy."

and we will be interested only in the Abelian case. Every simplicial group G is a Kan simplicial set [19, Theorem 17.1], and so continuous maps into $|G|$ have a simplicial representation up to homotopy.

A simplicial setting: twisted products. For our purposes, we will deal with fiber bundles where F, B, and E are simplicial sets, and a simplicial group G acts (simplicially) on F. The corresponding simplicial notion is called a *twisted Cartesian product* (a more general simplicial notion, a counterpart of a fibration, is a *Kan fibration*; see, e.g., [19, Chap. I,II]).

Definition 3.9 (Twisted Cartesian product). *Let B and F be simplicial sets, and let an action of a simplicial group G on F be given, i.e., a simplicial map $F \times G \to F$ satisfying the usual conditions for a (right) action of a group on a set; that is, $\phi(\gamma\gamma') = (\phi\gamma)\gamma'$ and $\phi e_k = \phi$ ($\phi \in F_k$, $\gamma, \gamma' \in G_k$, e_k the unit element of G_k). Moreover, let $\tau = (\tau_k)_{k=1}^{\infty}$ be a* twisting operator, *where $\tau_k \colon B_k \to G_{k-1}$ are mappings satisfying the following conditions (we omit the dimension indices for simplicity):*

(i) $\partial_0\tau(\beta) = \tau(\partial_1\beta)\tau(\partial_0\beta)^{-1}$;
(ii) $\partial_i\tau(\beta) = \tau(\partial_{i+1}\beta)$ *for* $i \geq 1$;
(iii) $s_i\tau(\beta) = \tau(s_{i+1}\beta)$ *for all* i*; and*
(iv) $\tau(s_0\beta) = e_k$ *for all* $\beta \in B_k$*, where* e_k *is the unit element of* G_k.

Then the twisted Cartesian product $F \times_\tau B$ *is a simplicial set E with $E_k = F_k \times B_k$, i.e., the k-simplices are as in the Cartesian product $F \times B$, and the face and degeneracy operators are also as in the Cartesian product (see Section 3.1), with the sole exception of ∂_0, which is given by*

$$\partial_0(\phi, \beta) := (\partial_0(\phi)\tau(\beta), \partial_0\beta), \qquad (\phi, \beta) \in F_k \times B_k.$$

A twisted Cartesian product $F \times_\tau B$ is called principal *if $F = G$ and the considered right action of G on itself is by (right) multiplication.*

Thus, the only way in which $F \times_\tau B$ differs from the ordinary Cartesian product $F \times B$ is in the 0th face operator. It is definitely not easy to see why this should be the right way of representing fiber bundles simplicially, but for us, it is only important that it works, and we will have explicit formulas available for the twisting operator for all the specific applications. Actually, we will use solely principal twisted Cartesian products.

Let F, B be locally polynomial-time simplicial sets, let G be a locally polynomial-time simplicial group, and let the action of G on F and the twisting operator τ be polynomial-time maps (again in a sense precisely analogous to polynomial-time simplicial maps or chain maps); we assume that all of these objects are parameterized by the same parameter set $\mathcal{I}$. It is easy to see that then the simplicial set $F \times_\tau B$, again parameterized by $\mathcal{I}$, is locally polynomial-time.

We will need that under certain reducedness assumptions, twisted products preserve polynomial-time homology.

Proposition 3.10 (Twisted product). *Let F and B be simplicial sets with polynomial-time homology, let G be a locally polynomial-time simplicial group with a polynomial-time simplicial action on F, and let τ be a polynomial-time twisting operator. Moreover, suppose that G is* 0-reduced *(a single vertex) or that B is* 1-reduced *(a single vertex, no edges). Then $E := F \times_\tau B$ can be equipped with polynomial-time homology.*

The effective-homology analogs of this result are due to Rubio and Sergeraert [35, Theorem 132] when B is 1-reduced and due to Filakovský [11, Corollary 12] when G is 0-reduced.

Proof (sketch). Let the polynomial-time homology of F and B be given by strong equivalences $C_*(F) \overset{\mathrm{P}}{\Longleftrightarrow} EF_*$ and $C_*(B) \overset{\mathrm{P}}{\Longleftrightarrow} EB_*$, respectively.

We begin with the ordinary Cartesian product $F \times B$. By the Eilenberg–Zilber theorem (Lemma 3.5 for two factors, where we do not need to assume 0-reducedness), there is a reduction $(\mathrm{AW}, \mathrm{EML}, \mathrm{SHI}) : C_*(F \times B) \overset{\mathrm{P}}{\Rightarrow} T_*$, where T_* is the tensor product $C_*(F) \otimes C_*(B)$. Further, by Lemma 3.4 for two factors, we have $T_* \overset{\mathrm{P}}{\Longleftrightarrow} ET_* := EF_* \otimes EB_*$. So altogether

$$C_*(F \times B) \overset{\mathrm{P}}{\Rightarrow} T_* \overset{\mathrm{P}}{\Longleftrightarrow} ET_*. \tag{5}$$

Next, by the definition of the twisted product, the chain complex $C_*(F \times_\tau B)$ has the same chain groups as $C_*(F \times B)$, but the differential is modified. Writing δ for the difference of the two differentials, on elements (ϕ, β) the standard basis of $C_k(F \times B)$ we get $\delta(\phi, \beta) = (\partial_0(\phi)\tau(\beta), \partial_0\beta) - (\partial_0\phi, \partial_0\beta)$.

We recall that in any simplicial set X, every simplex σ can be obtained from a unique *nondegenerate* simplex τ by an application of degeneracy operators. Let us refer to the dimension of τ as the *geometric dimension* of σ. Given a simplex (ϕ, β) of $F \times B$, its *filtration degree* is defined as the geometric dimension of β.

In the present proof, the filtration degree serves as a potential function for controlling nilpotency of the appropriate maps. First, it can be checked that the chain homotopy SHI does not increase the filtration degree, and a simple argument shows that δ decreases it at least by 1 (see, e.g., [35, Theorem 130], for details). It follows that the composition $\mathrm{SHI} \circ \delta$ has constant nilpotency bounds, namely, $N_k = k + 1$. Therefore, the basic perturbation lemma (Theorem 3.6) shows that $C_*(F \times_\tau B) \overset{\mathrm{P}}{\Rightarrow} T'_*$, where T'_* is a perturbation of the tensor product complex T_*.

Next, we would like to propagate the perturbation from T_* through the next strong equivalence in (5), which we write more explicitly as

$$T_* \overset{\mathrm{P}}{\Leftarrow} \hat{T}_* \overset{\mathrm{P}}{\Rightarrow} ET_*.$$

Let δ^T be the difference of the differential in T'_* and in T_*. By the easy perturbation lemma (Lemma 3.7), we get a perturbed version $\hat{T}'_*$ of the middle complex $\hat{T}_*$, and the difference of its differential minus the differential of $\hat{T}_*$ is $\hat{\delta}^T = g\delta^T f$, for some chain maps f, g from the reduction $T_* \overset{\mathrm{P}}{\Leftarrow} \hat{T}_*$.

We now recall from the proof of Lemma 3.4 that the chain complex $\hat{T}_*$ is constructed as a tensor product of two chain complexes, and that the chain homotopy h in the reduction $\hat{T}_* \overset{\mathrm{P}}{\Rightarrow} ET_*$ has the form

$$h = h^{(1)} \otimes \mathrm{id} + g^{(1)} f^{(1)} \otimes h^{(2)}, \tag{6}$$

for some chain maps $f^{(1)}, g^{(1)}$ and chain homotopies $h^{(1)}, h^{(2)}$.

In order to apply the basic perturbation lemma to the just mentioned reduction $\hat{T}_* \overset{\mathrm{P}}{\Rightarrow} ET_*$, we need to show that $h\hat{\delta}^T$ has constant nilpotency bounds for every chain homotopy h of the form (6). This follows from the obvious fact that such a chain homotopy never increases the filtration degree[21] by more than 1, plus a result showing that if G is 0-reduced or B is 1-reduced, then $\hat{\delta}^T$ *decreases* the filtration degree at least by 2. We refer to [11, Corollary 9

[21] For a basis element $\hat{a} \otimes \hat{b}$ of the tensor product $\hat{T}$, the filtration degree is defined simply as the degree of $\hat{b}$.

and 11] for a proof of the latter result (also see the proof of Lemma 3.14 below, where a very similar situation is discussed). Then a constant nilpotency bound with $N_k \leq k+1$ follows, and the proposition is proved. $\square$

3.5 The bar construction

The bar construction, originating in Eilenberg and Mac Lane [9], is an algebraic construction with many uses and generalizations. For us, it provides a way of constructing auxiliary chain complexes for certain reductions and strong equivalences; we will thus introduce it only in the setting of chain complexes. The definition below is somewhat complicated, but most of the details will be irrelevant in the sequel—the important properties will be encapsulated in a couple of lemmas below. We essentially follow [25, Chap. 3], with some minor technical differences.

A *differential graded algebra* is a chain complex A_* together with an associative multiplication $A_* \otimes A_* \to A_*$ with a unit 1_{A_*}. We denote the image of $a \otimes b$ simply by $a \cdot b$. This multiplication is assumed to be a chain map; in particular, for $a \in A_k$ and $b \in A_\ell$ we have $a \cdot b \in A_{k+\ell}$. The chain map condition on the multiplication reads

$$d(a \cdot b) = d(a) \cdot b + (-1)^{\deg a} a \cdot d(b)$$

(the Leibniz rule). The unit 1_{A_*} is necessarily of degree 0.

We say that A_* is 0-reduced if $A_0 = \mathbb{Z}$, generated by 1_{A_*}. Regarding $\mathbb{Z}$ as a chain complex whose all chain groups are zero except for the one in dimension 0, which is $\mathbb{Z}$, there is a unique homomorphism $\varepsilon\colon A_* \to \mathbb{Z}$ of differential graded algebras (i.e., a chain map preserving the unit and the multiplication).[22] We call ε the *augmentation*. Its kernel, the *augmentation ideal*, is denoted by $\overline{A}_*$.

Further, we denote by $\overline{A}^{\uparrow}_*$ the shift of $\overline{A}_*$ upwards by one, so that we have

$$\overline{A}^{\uparrow}_0 = \overline{A}^{\uparrow}_1 = 0, \text{ and } \overline{A}^{\uparrow}_k = A_{k-1}, k \geq 2.$$

The shifted chain complex comes with the shifted differential $d^{\overline{A}^{\uparrow}_*}(a) = -d^{\overline{A}_*}(a) = -d^{A_*}(a)$.

A right differential graded A_*-module is a chain complex M_* equipped with a chain map

$$M_* \otimes A_* \to M_*$$

that satisfies the usual axioms for a module structure. Again the action being a chain map translates into a Leibniz-type rule for the compatibility of the multiplication and the differential. Similarly, a left A_*-module N_* is equipped with an action $A_* \otimes N_* \to N_*$.

Given A_*, M_*, N_* as above, the bar construction produces a chain complex $\mathrm{Bar}^{A_*}(M_*, N_*)$. In order to define it, we first form an auxiliary chain complex given by

$$T_* := \bigoplus_{n=0}^{\infty} M_* \otimes (\overline{A}^{\uparrow}_*)^{\otimes n} \otimes N_*.$$

We denote the differential in T_* by d^T and call it the *tensorial differential*. The actual bar construction will be given by a perturbation of this differential.

[22] In detail $\varepsilon(n \cdot 1_{A_*}) = n$ and, for a of positive dimension, $\varepsilon(a) = 0$.

Assuming that each of the chain groups in A_*, M_*, N_* has a distinguished basis, the distinguished bases in T_* are made of elements of the form

$$z := x \otimes a_1 \otimes \cdots \otimes a_n \otimes y,$$

where x comes from a distinguished basis in M_*, y from one in N_*, and $a_1, \ldots, a_n \neq 1_A$ from those in A_*. (Here we can also explain the origin of the name "bar construction"; in the Eilenberg–Mac Lane founding paper, the tensor product signs $\otimes$ in the above notation for z were abbreviated to vertical bars.) The tensorial differential $d^T(z)$ is given by the (iterated) formula (3) from Section 3.1.

The degree of such a z equals $\deg(z) = \deg_{\text{tens}}(z) + \deg_{\text{res}}(z)$, where $\deg_{\text{tens}}(z)$, the *tensorial degree* of z, equals $\deg(x) + \deg(y) + \sum_{i=1}^n \deg(a_i)$ (with $\deg(a_i)$ being the degree of a_i in A_*), and the *residual degree* $\deg_{\text{res}}(z) = n$.

Now the chain complex $\mathrm{Bar}^{A_*}(M_*, N_*)$ has the same chain groups as T_*, but the differential is modified to $d^T + \delta^{\text{ext}}$, where δ^{ext}, the *external differential*, is given by

$$\begin{aligned}\delta^{\text{ext}}(x \otimes a_1 \otimes \cdots \otimes a_n \otimes y) \quad := \quad & (-1)^{m_0} x \cdot a_1 \otimes a_2 \otimes \cdots \otimes a_n \otimes y \\ & + \sum_{i=1}^{n-1} (-1)^{m_i} x \otimes a_1 \otimes \cdots \otimes a_i \cdot a_{i+1} \otimes \cdots \otimes a_n \otimes y \\ & + (-1)^{m_n} x \otimes a_1 \otimes \cdots \otimes a_{n-1} \otimes a_n \cdot y,\end{aligned}$$

where $m_i = \deg(x) + \deg(a_1) + \cdots + \deg(a_i) + i$. We note that the external differential is the only part of the definition of $\mathrm{Bar}^{A_*}(M_*, N_*)$ where the algebra and module structures play a role. This finishes the definition of the bar construction.

In our applications, the bar construction will be used with M_* equal to $\mathbb{Z}$. Here we endow $\mathbb{Z}$ with the right A_*-module structure obtained from the augmentation—the unit 1_{A_*} acts by identity as it must and the elements from the augmentation ideal act trivially, i.e., $a \cdot x = 0$. We also note that $\mathbb{Z}$ acts as a unit element for tensor product, in the sense that $C_* \otimes \mathbb{Z}$ and $\mathbb{Z} \otimes C_*$ can be canonically identified with C_* (this is obvious by considering the distinguished bases, for example).

Lemma 3.11 (Polynomial-time homology for the bar construction)**.** *Let A_*, M_*, N_* be locally polynomial-time versions of the objects above, with all the multiplications involved being polynomial-time maps, and let us suppose that A_*, M_*, N_* are equipped with polynomial-time homology. Then* $\mathrm{Bar}^{A_*}(M_*, N_*)$ *can be equipped with polynomial-time homology.*

Proof. First we equip T_* with polynomial-time homology; this is essentially Lemma 3.4 about tensor products of strong equivalences. The factors M_* and N_* are not 0-reduced but this can be accommodated, in a way similar to Cartesian products—see the remark following Proposition 3.2. We also note that although T_* is an infinite direct sum, the kth chain group involves only elements with $n \leq k$ from this direct sum, and so Lemma 3.4 is applicable.

Next, we apply the easy perturbation lemma and the basic one, in a way very similar to the proof of Proposition 3.10 on twisted products, to propagate the perturbation of the differential in T_* by the external differential δ^{ext} through the strong equivalence. The only issue is to show constant nilpotency bounds. Here one uses that the chain homotopy involved, which is of the form (6) but with an arbitrary number of factors, does not increase the residual degree $\deg_{\text{res}}$, while δ^{ext}, obviously, decreases it by 1. $\square$

The next lemma is a key property of the bar construction, showing that it provides, in a sense, an "inverse" to the operation of tensor product with A_*. Indeed, the bar construction $\mathrm{Bar}^{A_*}(\mathbb{Z}, A_*)$ can be regarded as a formal analog of the power series expression $1 = \frac{a}{a} = a + (1-a)a + (1-a)^2 a + \cdots$ for a real number $a \in (0, 2)$.

Lemma 3.12. *Given a locally polynomial-time* 0*-reduced differential graded algebra A_*, there is a reduction*

$$\mathrm{Bar}^{A_*}(\mathbb{Z}, A_*) \overset{\mathrm{P}}{\Rrightarrow} \mathbb{Z}$$

(where A_* is taken as a differential graded A_*-module in the obvious way). More generally, if we consider, in addition, a locally polynomial-time chain complex M_* and turn $A_* \otimes M_*$ into a left A_*-module by defining $a \cdot (b \otimes x) := (a \cdot b) \otimes x$, then we obtain a reduction $\mathrm{Bar}^{A_*}(\mathbb{Z}, A_* \otimes M_*) \overset{\mathrm{P}}{\Rrightarrow} M_*$.

We note that we assume no A_*-module structure on M_*; the left A_*-module structure on $A_* \otimes M_*$ comes from the multiplication in A_*.

Proof. In the reduction $(f, g, h)\colon \mathrm{Bar}^{A_*}(\mathbb{Z}, A_*) \overset{\mathrm{P}}{\Rrightarrow} \mathbb{Z}$, f and g are given by the assumed identification of A_0 with $\mathbb{Z}$ (note that the 0th chain group of $\mathrm{Bar}^{A_*}(\mathbb{Z}, A_*)$ can be canonically identified with A_0); in particular, we have $f(a_1 \otimes \cdots \otimes a_n \otimes a) = 0$ unless $n = 0$.

In residual degree 0 we have $f(a) = \varepsilon(a)$. Denote by $\overline{a} = a - \varepsilon(a) \cdot 1_A$ the projection of a onto the augmentation ideal $\overline{A}_*$. Then, for a basis element $z = a_1 \otimes \cdots \otimes a_n \otimes a$ of $\mathrm{Bar}^{A_*}(\mathbb{Z}, A_*)$, we put

$$h(z) := (-1)^{\deg(a_1)+\cdots+\deg(a_n)+\deg(a)+n+1} a_1 \otimes \cdots \otimes a_n \otimes \overline{a} \otimes 1_{A_*}.$$

It is simple to check that we indeed get a reduction (see [20]), and polynomiality is obvious.

The more general reduction $\mathrm{Bar}^{A_*}(\mathbb{Z}, A_* \otimes M_*) \overset{\mathrm{P}}{\Rrightarrow} M_*$ is then immediately obtained from the previous one by tensoring all the maps with the identity on M_*. □

3.6 The base space (a "twisted division")

Here, as in Section 3.4, G is an Abelian simplicial group, and we consider a twisted product, this time a principal one: $G \times_\tau B$. However, while previously we took G, B, τ as known, and wanted to compute $G \times_\tau B$ (so we did "twisted multiplication"), here we assume that G *and* $G \times_\tau B$ are known, and we want B—so one can think of this as "twisted division". The bar construction is the main tool.

Proposition 3.13. *Let G be a* 0*-reduced locally polynomial-time Abelian simplicial group, let B be a locally polynomial-time simplicial set, and let τ be a polynomial-time twisting operator. If both G and $G \times_\tau B$ are equipped with polynomial-time homology, then B can also be equipped with polynomial-time homology.*

Proof. We follow the treatment in Real [25]. We let $A_* := C_*(G)$ be the normalized chain complex of G. The *Eilenberg–MacLane product* on A_* is defined using the operator $\mathrm{EML}\colon A_* \otimes A_* \to C_*(G \times G)$ as in the proof of Lemma 3.5. Writing $\mathrm{EML}(a \otimes b) = \sum_{i=1}^{n} \alpha_i (\gamma_i, \gamma_i')$, $\gamma_1, \ldots, \gamma_n' \in G$, we set

$$a \cdot b := \sum_{i=1}^{n} \alpha_i \gamma_i \gamma_i',$$

where $\gamma_i \gamma_i'$ is computed using the group operation in G. This multiplication is polynomial-time, and with some work it can be checked that it makes A_* into a differential graded algebra.

The untwisted case. First we assume that the ordinary Cartesian product $G \times B$ is given with polynomial-time homology. Then polynomial-time homology for B is obtained in the following steps:

1. $C_*(G \times B)$ has polynomial-time homology by the assumption.

2. The Eilenberg–Zilber reduction $C_*(G \times B) \overset{\mathrm{P}}{\Rrightarrow} A_* \otimes C_*(B)$ (Lemma 3.5) and the composition of strong equivalences yield polynomial-time homology for $A_* \otimes C_*(B)$.

3. Since A_* has polynomial-time homology as well by assumption, Lemma 3.11 yields polynomial-time homology for $\mathrm{Bar}^{A_*}(\mathbb{Z}, A_* \otimes C_*(B))$.

4. Finally, the reduction $\mathrm{Bar}^{A_*}(\mathbb{Z}, A_* \otimes C_*(B)) \overset{\mathrm{P}}{\Rrightarrow} C_*(B)$ from Lemma 3.12 and composition of strong equivalences provide polynomial-time homology for $C_*(B)$.

The twisting. Now we present "twisted analogs" of steps 1–4 above.

1_τ. We assume that polynomial-time homology is available for the *twisted* Cartesian product $G \times_\tau B$.

2_τ. As in the proof of Proposition 3.10 (twisted product), applying the basic perturbation lemma to the Eilenberg–Zilber reduction $C_*(G \times B) \overset{\mathrm{P}}{\Rrightarrow} Q_* := A_* \otimes C_*(B)$ provides a reduction $C_*(G \times_\tau B) \overset{\mathrm{P}}{\Rrightarrow} Q'_*$, where Q'_* is obtained by perturbing the differential d^Q of the tensor product complex Q_* to another differential $d^{Q'}$. Let $\delta^Q := d^{Q'} - d^Q$ be the difference. On Q'_* the multiplication by A_* from the left is defined in the same way as on Q_*. Using formula (7) below, one can prove that the perturbation δ^Q is A_*-linear. It means that $d^{Q'}$ satisfies the Leibniz rule and hence Q'_* is a left A_*-module.

3_τ. We have $d^{Q'}$ polynomial-time computable (since the basic perturbation lemma provides an explicit formula), and hence we obtain polynomial-time homology for $\mathrm{Bar}^{A_*}(\mathbb{Z}, Q'_*)$ by Lemma 3.11.

4_τ. It remains to exhibit a reduction $\mathrm{Bar}^{A_*}(\mathbb{Z}, Q'_*) \overset{\mathrm{P}}{\Rrightarrow} C_*(B)$; then we obtain polynomial-time homology for B as in the untwisted case above. We begin with the reduction $\mathrm{Bar}^{A_*}(\mathbb{Z}, Q_*) \overset{\mathrm{P}}{\Rrightarrow} C_*(B)$ from Lemma 3.12 and apply the basic perturbation lemma to it.

We note that, by the definition of the bar construction, $\mathrm{Bar}^{A_*}(\mathbb{Z}, Q_*)$ and $\mathrm{Bar}^{A_*}(\mathbb{Z}, Q'_*)$ have the same chain groups, and only the differential is modified. Let δ^{Bar} be the differential of $\mathrm{Bar}^{A_*}(\mathbb{Z}, Q'_*)$ minus the one of $\mathrm{Bar}^{A_*}(\mathbb{Z}, Q_*)$. We observe that the external differentials in these bar constructions coincide, and the tensorial differentials differ only in one term. Thus, writing a basis element of $\mathrm{Bar}^{A_*}(\mathbb{Z}, Q_*)$ as $z = a_1 \otimes \cdots \otimes a_n \otimes (a \otimes b)$, we have

$$\delta^{\mathrm{Bar}} z = (-1)^{\deg(a_1)+\cdots+\deg(a_n)+\deg(a)-n} a_1 \otimes \cdots \otimes a_n \otimes \delta^Q(a \otimes b).$$

The rest of the proof is delegated to the next lemma, which is essentially Prop. 3.2.3 in [25].

□

Lemma 3.14. *If G is a 0-reduced simplicial group, $A_* = C_*(G)$ and $Q_* = A_* \otimes C_*(B)$ are as above, $(f, g, h)\colon \mathrm{Bar}^{A_*}(\mathbb{Z}, Q_*) \Rrightarrow C_*(B)$ is the reduction from Lemma 3.12, and δ^{Bar} is the perturbation of the differential of $\mathrm{Bar}^{A_*}(\mathbb{Z}, Q_*)$ as above, then $h\delta^{\mathrm{Bar}}$ has constant nilpotency bounds, and the perturbed differential in $C_*(B)$ obtained from the application of the basic perturbation lemma to the reduction (f, g, h) actually equals the original differential in $C_*(B)$, i.e., the resulting perturbation is zero.*

Proof. There is an explicit expression known for the perturbation δ^Q, going back to Brown [4] and Shih [40]. We do not need the full explicit formula, just some of its properties.

Namely, given G, B, and the twisting operator τ, there is a sequence of homomorphisms $t_k\colon C_k(B) \to C_{k-1}(G)$, such that for $a \in C_\ell(G)$, $b \in C_k(B)$, we have

$$\delta^Q(a \otimes b) = \sum_{i=0}^{k} (-1)^\ell a \cdot t_{k-i}(b_{k-i}) \otimes \tilde{b}_i, \tag{7}$$

for some chains $b_0, \ldots, b_k, \tilde{b}_0, \ldots, \tilde{b}_k$, with $b_i, \tilde{b}_i \in C_i(B)$, the multiplication in $a \cdot t_{k-i}(b_{k-i})$ being the Eilenberg–MacLane product introduced above.[23]

Now $t_0 = 0$ since $C_{-1}(G) = 0$. Moreover, one can compute (see the proof of [11, Corollary 11]) that $t(b_1) = \tau(b_1) - e_0$ for all 1-simplices $b_1 \in B_1$. Since G is 0-reduced, it follows that $t_1 = 0$.

Hence the sum in (7) goes only up to $i = k - 2$, and so δ^Q decreases the filtration degree (given by the degree in $C_*(B)$) at least by 2. The same applies to δ^{Bar} when we take the filtration on $\mathrm{Bar}^{A_*}(\mathbb{Z}, Q_*)$ given again by the degree in $C_*(B)$. Similar to the conclusion of the proof of Proposition 3.10, we obtain constant nilpotency bound of $h\delta^{\mathrm{Bar}}$.

It remains to show that the perturbation of the differential in $C_*(B)$ obtained by using the basic perturbation lemma to the reduction $(f, g, h)\colon \mathrm{Bar}^{A_*}(\mathbb{Z}, Q_*) \Rrightarrow C_*(B)$ with the perturbation δ^{Bar} is zero. As was mentioned in connection with the basic perturbation lemma, the considered perturbation equals $f\delta^{\mathrm{Bar}}\varphi g$, where $\varphi = \sum_{i=0}^{\infty} (-1)^i (h\delta^{\mathrm{Bar}})^i$.

We will check that $f\delta^{\mathrm{Bar}} = 0$. Indeed, the mapping f in the reduction from Lemma 3.12 is obtained from the augmentation $\varepsilon\colon A_* \to \mathbb{Z}$ by tensoring with $\mathrm{id}_{C_*(B)}$. Thus, if $z = a_1 \otimes \cdots \otimes a_n \otimes (a \otimes b)$ is a basis element, we have $f(z) = 0$ unless $\deg(a) = 0$. But the Eilenberg–MacLane product $a \cdot t_{k-i}(b_{k-i})$ in the formula (7) has degree at least $\deg(t_{k-i}(b_{k-i})) = k-i-1$. Thus, the degree can be 0 only for $k-i-1 = 0$, but in this case $t_{k-i} = t_1 = 0$, and so $f\delta^{\mathrm{Bar}} = 0$ as claimed. □

3.7 Eilenberg–MacLane spaces

Preliminaries on cochains. Before entering the realm of Eilenberg–MacLane spaces, we recall a few notions related to cohomology. Throughout this section, let π be an Abelian group.

For us, it will often be convenient to regard cochains as homomorphisms from chain groups into π. That is, given a chain complex C_* (whose chain groups are, as always in this paper,

[23] In the literature, t is called a *twisting cochain*, and $\delta^Q(a \otimes b)$ is written as a cap product $t \cap (a \otimes b)$. Moreover, t is in general not determined uniquely by G, B, τ, since the operator AW in the reduction $C_*(G \times B) \Rrightarrow C_*(G) \otimes C_*(B)$ is not unique. However, the relevant sources use the same particular AW as we do.

free Abelian groups), we define its *kth cochain group with coefficients in* π as $C^k(C_*;\pi) := \mathrm{Hom}(C_k,\pi)$, with pointwise addition. The *coboundary operator* $\delta_k\colon C^k(C_*;\pi)\to C^{k+1}(C_*;\pi)$ is then given by $(\delta_k c^k)(c_{k+1}) := c^k(d_{k+1}c_{k+1})$ for every k-cochain c^k and every $(k+1)$-chain c_{k+1}.[24] (The notation δ was earlier used for a perturbation of a differential, but from now on, we will encounter it only in the role of a coboundary operator.)

In particular, if X is a simplicial set, the *normalized cochain complex* $C^*(X;\pi)$ is $C^*(C_*(X);\pi)$; thus, a k-cochain can be specified by its values on the standard basis, i.e., as a labeling of the nondegenerate k-simplices by elements of π—this agrees with the usual definition in introductory textbooks.

For us, it will be important that if X has infinitely many nondegenerate k-simplices, then a k-cochain in $C^k(X)$ is an infinite object (unlike a k-chain!). Thus, in algorithms, we will need to use a black-box representation of individual cochains—the black box supplies the value of the cochain on a given simplex (or on a given chain, which is computationally equivalent).

To finish our remark on cochains, we recall that if C^* is a cochain complex, with coboundary operator $\delta=(\delta_k)_{k\in\mathbb{Z}}$, then $B^k := \mathrm{im}\,\delta_{k-1}$ is the group of *k-coboundaries*, $Z^k := \ker\delta_k$ the group of *k-cocycles*, and $H^k = H^k(C^*;\pi) := Z^k/B^k$ is the *kth cohomology group*.

Eilenberg–MacLane spaces topologically. For an Abelian group π and an integer $k\geq 1$, the *Eilenberg–MacLane space* $K(\pi,k)$ is defined as any topological space T with $\pi_k(Z)\cong\pi$ and $\pi_i(T)=0$ for all $i\neq k$ (actually, $K(\pi,1)$ is also defined for an arbitrary group π, but we will consider solely the Abelian case).

It is known, and not too hard to prove, that a $K(\pi,k)$ exists for all $k\geq 1$ and all π, and it is also known to be unique up to homotopy equivalence.[25]

The definition postulates that the homotopy groups of an Eilenberg–MacLane space are, in a sense, the simplest possible, and this makes it relatively easy to understand the structure of all maps from a given space X into $K(\pi,k)$. Indeed, a basic topological result says that

$$[X,K(\pi,k)]\cong H^k(X;\pi), \tag{8}$$

assuming that X is a “reasonable” space (say a CW-complex). In words, homotopy classes of maps $X\to K(\pi,k)$ correspond to the elements of the kth cohomology group of X with coefficients in π (see, e.g., [19, Lemma 24.4] for this fact in a simplicial setting, and [6] for a geometric explanation).

The standard simplicial model. There is a standard way of representing $K(\pi,k)$ as a Kan simplicial set, which actually is even a simplicial group. We will work with this simplicial representation, and from now on, the notation $K(\pi,k)$ will be reserved for this particular simplicial representation, to be defined next.

Let Δ^ℓ denote the ℓ-dimensional standard simplex, regarded as a simplicial complex (or a simplicial set; the difference is purely formal in this case). That is, the vertex set is $\{0,1,\ldots,\ell\}$ and the k-dimensional (nondegenerate) simplices are all $(k+1)$-element subsets of $\{0,1,\ldots,\ell\}$.

The set of ℓ-simplices of $K(\pi,k)$ is given by

$$K(\pi,k)_\ell := Z^k(\Delta^\ell;\pi);$$

[24] Sometimes other conventions are used for the coboundary operator in the literature; e.g. $(\delta_k c^k)(c_{k+1}) = (-1)^{k+1}c^k(d_{k+1}c_{k+1})$. But our main sources [19] and [15] use the version without signs.

[25] Provided that we restrict to spaces that are homotopy equivalent to CW-complexes.

that is, each ℓ-simplex is (represented by) a k-dimensional cocycle on Δ^ℓ. Thus, it can be regarded as a labeling of the k-dimensional faces of Δ^ℓ by elements of the group π; moreover, the labels must add up to 0 on the boundary of every $(k+1)$-face.

It is also easy to define the face and degeneracy operators in $K(\pi,k)$. Given an ℓ-simplex σ of $K(\pi,k)$, represented as a labeling of the k-faces of Δ^ℓ, $\partial_i\sigma$ is defined as the restriction of σ on the ith $(\ell-1)$-face of Δ^ℓ. (The ith $(\ell-1)$-face of Δ^ℓ is identified with $\Delta^{\ell-1}$ via the unique order-preserving bijection of the vertex sets.) As for the degeneracy operators, $s_i\sigma$ is the labeling of k-faces of $\Delta^{\ell+1}$ induced by the mapping $\eta_i\colon \{0,1,\ldots,\ell+1\} \to \{0,1,\ldots,\ell\}$ given by

$$\eta_i(j) = \begin{cases} j & \text{for } j \le i, \\ j-1 & \text{for } j > i. \end{cases}$$

In particular, if a k-face contains both i and $i+1$, then it is labeled by 0, since its η_i-image is a degenerate simplex.

The simplicial group operation in $K(\pi,k)$ is the addition of cocycles in $Z^k(\Delta^\ell;\pi)$.

In the simplicial setting we have

$$\mathrm{SMap}(X, K(\pi,k)) \cong Z^k(X;\pi) \tag{9}$$

for every simplicial set X. That is, simplicial maps $X \to K(\pi,k)$ are in a bijective correspondence with π-valued k-cocycles on X (see below for an explicit description of this correspondence). Moreover, two such simplicial maps, represented by cocycles z and z', are homotopic iff $z - z'$ is a coboundary (see, e.g., [19, Theorem 24.4]). This immediately implies $[X, K(\pi,k)] \cong H^k(X;\pi)$, which was mentioned above in (8).

The set $\boldsymbol{E(\pi,k)}$. In addition to the simplicial Eilenberg–MacLane space $K(\pi,k)$ we also need another simplicial set, denoted by $E(\pi,k)$. While the ℓ-simplices of $K(\pi,k)$ are all k-cocycles on Δ^ℓ, the ℓ-simplices of $E(\pi,k)$ are all k-cochains:

$$E(\pi,k)_\ell := C^k(\Delta^\ell;\pi).$$

The face and degeneracy operators are defined in exactly the same way as those of $K(\pi,k)$.

Converting between simplicial maps and cochains. We have mentioned that simplicial maps $X \to K(\pi,k)$ are in one-to-one correspondence with cocycles in $Z^k(X;\pi)$. Similarly, simplicial maps $X \to E(\pi,k)$ correspond to cochains in $C^k(X;\pi)$:

$$\mathrm{SMap}(X, E(\pi,k)) \cong C^k(X;\pi).$$

Let us describe this correspondence explicitly, since we will need it in the algorithm. First we note that a k-simplex τ of $E(\pi,k)$ is a k-cochain on Δ^k, i.e., a labeling of the single k-face of Δ^k by an element of π. Let us denote this element by $\mathrm{ev}(\tau)$ (here ev stands for "evaluation").

Given a simplicial map $f\colon X \to E(\pi,k)$, the corresponding cochain $\kappa \in C^k(X;\pi)$ is simply given by $\kappa(\sigma) = \mathrm{ev}(f(\sigma))$ for every $\sigma \in X_k$ (where on the left-hand side, σ is taken as a generator of the chain group $C_k(X)$).

Conversely, given $\kappa \in C^k(X;\pi)$, we describe the corresponding simplicial map f. The value $f(\sigma)$ on an ℓ-simplex $\sigma \in X_k$ should be a k-chain on Δ^ℓ. There is a unique simplicial map $i_\sigma\colon \Delta^\ell \to X$ that sends the nondegenerate ℓ-simplex of Δ^ℓ to σ (indeed, a simplicial map has to respect the ordering of vertices, implicit in the face and degeneracy operators). Then $f(\sigma)$ is the cochain $i_\sigma^*(\kappa)$, i.e., the labels of the k-faces of σ given by κ are pulled back to Δ^ℓ. Moreover, if κ is a cocycle, then f goes into $K(\pi,k)$.

A useful fibration. Since an ℓ-simplex $\sigma \in E(\pi,k)$ is formally a k-cochain, we can take its coboundary $\delta\sigma$. This is a $(k+1)$-coboundary (and thus also cocycle), which we can interpret as an ℓ-simplex of $K(\pi,k+1)$. It turns out that this induces a *simplicial* map $E(\pi,k) \to K(\pi,k+1)$, which is (with the usual abuse of notation) also denoted by δ. This map is actually surjective, since the relevant cohomology groups of Δ^ℓ are all zero and thus all cocycles are also coboundaries.

As is well known, $\delta\colon E(\pi,k) \to K(\pi,k+1)$ is a fiber bundle with fiber $K(\pi,k)$.

There is another simplicial description of $E(\pi,k)$ as a twisted product

$$K(\pi,k) \times_\tau K(\pi,k+1),$$

where τ has the following explicit form (see [19, §23] or [35, Sec. 7.10.2]):

Let $z \in Z^{k+1}(\Delta^\ell;\pi)$ be an ℓ-simplex of $K(\pi,k+1)$, i.e., a labeling of the $(k+1)$-faces of Δ^ℓ by elements of π (satisfying the cocycle condition). Then we want $\tau(z)$ to be an $(\ell-1)$-simplex of $K(\pi,k)$, i.e., a labeling of k-faces of $\Delta^{\ell-1}$. If we write a k-face of $\Delta^{\ell-1}$ as an increasing $(k+1)$-tuple $(i_0,\ldots,i_k)$, $0 \le i_0 < \cdots < i_k \le \ell-1$, we set

$$(\tau(z))(i_0,\ldots,i_k) := z(0,i_0+1,i_1+1,\ldots,i_k+1) - z(1,i_0+1,i_1+1,\ldots,i_k+1). \qquad (10)$$

The twisted product $K(\pi,k) \times_\tau K(\pi,k+1)$ is simplicially isomorphic to $E(\pi,k)$ as defined earlier. The isomorphism will be described, in a slightly more general setting, in the proof of Corollary 3.18 below.

3.8 Polynomial-time homology for $K(\pi,k)$

A crucial ingredient in our algorithm for computing Postnikov systems is obtaining polynomial-time homology for $K(\pi,k)$. Here, as usual, we assume k fixed, and π is a globally polynomial-time Abelian group (as introduced after Definition 2.3); then $K(\pi,k)$ has the same parameter set as π. It is easily checked that $K(\pi,k)$ is a locally polynomial-time simplicial group.

The $\overline{W}$ construction. Polynomial-time homology for $K(\pi,k)$ will be constructed by induction on k. The inductive step is based on a construction $\overline{W}$ (see [19, pages 87–88]) that, given an Abelian simplicial group G, produces another Abelian simplicial group $\overline{W}G$. The k-simplices have the form $\omega = (\gamma_{k-1},\gamma_{k-2},\ldots,\gamma_0)$, where γ_i is an i-simplex of G, $i = 0,1,\ldots,k-1$, and the group operation in $\overline{W}G$ is obtained by using the operation of G componentwise. The face operators are

$$\begin{aligned}
\partial_0\omega &:= (\gamma_{k-2},\gamma_{k-3},\ldots,\gamma_0),\\
\partial_{i+1}\omega &:= (\partial_i\gamma_{k-1},\ldots,\partial_1\gamma_{k-i},\underbrace{\partial_0\gamma_{k-i-1}+\gamma_{k-i-2}}_{\text{operation in } G},\gamma_{k-i-3},\ldots,\gamma_0), \quad i = 0,1,\ldots,k-1,
\end{aligned}$$

and the degeneracy operators are given by

$$\begin{aligned}
s_0\omega &:= (e_k,\gamma_{k-1},\ldots,\gamma_0),\\
s_{i+1}\omega &:= (s_i\gamma_{k-1},\ldots,s_0\gamma_{k-i-1},e_{k-i-1},\gamma_{k-i-2},\ldots,\gamma_0), \quad i = 0,1,\ldots,k-1,
\end{aligned}$$

where e_k is the unit element of G_k.

Topologically, $\overline{W}G$ is the classifying space of G, usually denoted by BG, but we won't use this fact directly. What we need is the following simplicial isomorphism.

Lemma 3.15. *For every Abelian group π and every $k \geq 1$, there is a simplicial isomorphism*

$$f\colon K(\pi, k+1) \to \overline{W}K(\pi, k);$$

if k is fixed and π is globally polynomial-time, then both f and f^{-1} are polynomial-time maps. Consequently, polynomial-time homology for $\overline{W}K(\pi, k)$ yields polynomial-time homology for $K(\pi, k+1)$.

Proof. We define an auxiliary simplicial set $WK(\pi, k)$ as the twisted Cartesian product $K(\pi, k) \times_\tau \overline{W}K(\pi, k)$, where $\tau\colon K(\pi, k+1) \to K(\pi, k)$ is the twisting operator of δ introduced at the end of Section 3.7. Then, according to [19, Theorem 23.10], there are simplicial isomorphisms $f\colon K(\pi, k+1) \to \overline{W}K(\pi, k)$ and $F\colon E(\pi, k) \to WK(\pi, k)$ that are compatible with respect to the projection maps $\delta\colon E(\pi, k) \to K(\pi, k+1)$ and $WK(\pi, k) \to \overline{W}K(\pi, k)$. By [19, Lemma 21.9] and the formula (1) there, the isomorphism f maps $z \in K(\pi, k+1)_\ell$ to

$$f(z) := \Big(\tau(z), \tau(\partial_0 z), \tau(\partial_0^2 z), \ldots, \tau(\partial_0^{\ell-1} z)\Big) \in \overline{W}K(\pi, k)_\ell$$

where τ is the twisting operator as above. Combining these statements together it follows that f is an isomorphism, and to finish the proof, we need to compute its inverse in polynomial time.

We describe an inductive algorithm for this. First we note that

$$f(z) = (\tau(z), f(\partial_0 z)).$$

There is only one simplex in dimension at most k in both of the considered simplicial sets, so the isomorphism is given uniquely there. A $(k+1)$-simplex of $\overline{W}K(\pi, k)$ has the form $\omega = (w_k, 0, 0, \ldots, 0)$, where $w_k \in Z^k(\Delta^k; \pi)$. Defining $z_{k+1} \in K(\pi, k+1)_{k+1} = Z^{k+1}(\Delta^{k+1}; \pi)$ by $z_{k+1}(0, 1, 2, \ldots, k+1) := w_k(0, 1, \ldots, k)$, we get $f(z_{k+1}) = (\tau(z_{k+1}), 0, \ldots, 0) = \omega$, so we have found $f^{-1}(\omega)$.

Next, we suppose that we can compute f^{-1} for simplices up to dimension $\ell \geq k+1$, and let $\omega = (w_\ell, w_{\ell-1}, \ldots, w_0) \in \overline{W}K(\pi, n)_{\ell+1}$. In order to obtain $z = f^{-1}(\omega)$, we first inductively compute $z' = f^{-1}(w_{\ell-1}, \ldots, w_0)$; then $z' = \partial_0 z$, and by the definition of ∂_0 in $K(\pi, k+1)$, we get that for $1 \leq i_0 < i_1 < \cdots < i_{k+1} \leq \ell + 1$ we have

$$z(i_0, i_1, \ldots, i_{k+1}) = z'(i_0 - 1, i_1 - 1, \ldots, i_{k+1} - 1). \tag{11}$$

On the other hand, for $0 = i_0 < i_1 < \cdots < i_{k+1} \leq \ell + 1$, from the formula (10) defining τ we obtain

$$\begin{aligned} \tau(z)(i_1 - 1, \ldots, i_{k+1} - 1) &= z(0, i_1, \ldots, i_{k+1}) - z(1, i_1, \ldots, i_{k+1}) \\ &= z(0, i_1, \ldots, i_{k+1}) - z'(0, i_1 - 1, \ldots, i_{k+1} - 1). \end{aligned} \tag{12}$$

From this we can express $z(0, i_1, \ldots, i_k)$ in terms of $\tau(z) = w_\ell$ and z', which are both known. This finishes the construction of the inverse. □

Now we can state the main result of this section.

Theorem 3.16. *Let $k \geq 1$ be a fixed integer. The standard simplicial model of the Eilenberg–MacLane space $K(\pi, k)$, where π is a globally polynomial-time Abelian group, can be equipped with polynomial-time homology.*

Proof. The proof proceeds by induction on k. The base case is $K(\pi,1)$, and it goes as follows.

1. Polynomial-time homology for $K(\mathbb{Z},1)$ is the main result of [18].
2. Polynomial-time homology for $K(\mathbb{Z}/m,1)$ is derived from that for $K(\mathbb{Z},1)$ in Lemma 3.17 below.
3. For π arbitrary, we use the specified polynomial-time isomorphism $\pi \cong \mathrm{Ab}(\mathbf{m})$ to write $K(\pi,1) \cong K(\mathrm{Ab}(\mathbf{m}),1)$. Since $\mathrm{Ab}(\mathbf{m})$ decomposes into a direct sum of cyclic groups, we can obtain polynomial-time homology for $K(\pi,1)$ using
$$K(\pi_1 \oplus \cdots \oplus \pi_s, 1) \cong K(\pi_1,1) \times \cdots \times K(\pi_s,1),$$
which is easy to see from the definition of $K(\pi,1)$, plus Proposition 3.2 (product with many factors).

The inductive step from $K(\pi,k)$ to $K(\pi,k+1)$ is as in [25], and it goes as follows.

1. To get polynomial-time homology for $K(\pi,k+1)$, according to Lemma 3.15 it suffices to obtain polynomial-time homology for $\overline{W}K(\pi,k)$.
2. With $G = K(\pi,k)$, let us consider the twisted product $G \times_\tau \overline{W}G$, where the twisting operator is given by $\tau_\ell(\gamma_{\ell-1},\dots,\gamma_0) := \gamma_{\ell-1}$ (this twisted product was denoted by WG in the proof of Lemma 3.15). Then there is a reduction
$$(f,g,h) : C_*(G \times_\tau \overline{W}G) \overset{\mathrm{P}}{\Rightarrow} \mathbb{Z},$$
with f,g defined in the obvious way (note that both G and $\overline{W}G$ are 0-reduced), and with h given by $h_\ell(\gamma_\ell,(\gamma_{\ell-1},\dots,\gamma_0)) := (e_{\ell+1},(\gamma_\ell,\gamma_{\ell-1},\dots,\gamma_0))$, where $e_{\ell+1}$ is the unit element of $G_{\ell+1}$ (see [19, page 88]). Thus, using Proposition 3.13 (twisted division) with $B = \overline{W}G$, we obtain polynomial-time homology for $\overline{W}G$ from that of G.

The proof of Theorem 3.16 is finished, except for the proof of the next lemma. □

Lemma 3.17. *Given a polynomial-time homology for $K(\mathbb{Z},1)$, one can equip $K(\mathbb{Z}/m,1)$ (parameterized by the natural number m encoded in binary) with polynomial-time homology.*

We note that the simplicial set $K(\mathbb{Z}/m,1)$ has finitely many simplices in each dimension (the number is even bounded by a polynomial in m for every fixed dimension). Nevertheless, we cannot treat it as a finite simplicial set, since it is parameterized by the group $\mathbb{Z}/m$, whose encoding size is only $\log m$, and so the number of simplices is exponential in this size. Somewhat paradoxically, we will use the infinite simplicial set $K(\mathbb{Z},1)$ to get a handle on the finite (in every dimension) $K(\mathbb{Z}/m,1)$.

Proof. By the assumption, the simplicial group $K(\mathbb{Z},1)$ is equipped with polynomial-time homology.

We will exhibit a twisting operator τ such that the principal twisted Cartesian product $P := K(\mathbb{Z},1) \times_\tau K(\mathbb{Z}/m,1)$ is simplicially isomorphic to $K(\mathbb{Z},1)$. Let $\varphi\colon P \to K(\mathbb{Z},1)$ be the isomorphism; assuming that both φ and φ^{-1} are polynomial-time maps, we can thus equip P with polynomial-time homology as well. Then we obtain the desired polynomial-time homology for $K(\mathbb{Z}/m,1)$ from Proposition 3.13 (twisted division).

Conceptually, the isomorphism φ is obtained from the short exact sequence of Abelian groups

$$0 \longrightarrow \mathbb{Z} \xrightarrow{\times m} \mathbb{Z} \xrightarrow{\text{mod } m} \mathbb{Z}/m \longrightarrow 0$$

by passing to classifying spaces. But our presentation below does not refer to this approach and is completely elementary.

In order to define φ and τ, it will be convenient to use a particular representation of simplices in $K(\mathbb{Z}, 1)$ and in $K(\mathbb{Z}/m, 1)$, described next.

We recall that the ℓ-simplices of $K(\mathbb{Z}, 1)$ are 1-dimensional integral cocycles on Δ^ℓ, in other words, labelings c of the edges of the complete graph on $\{0, 1, \dots, \ell\}$ with integers such that, for every triple $i < j < k$, $c(i, j) - c(i, k) + c(j, k) = 0$. It is easy to see that every such labeling is determined by a "potential function" a on the vertex set, i.e., $c(i, j) = a(j) - a(i)$ (from the topological point of view, every cocycle c is a coboundary since Δ^ℓ is contractible, and a is a 0-cochain with $c = \delta a$). Moreover, w.l.o.g. we can assume that $a(0) = 0$, and then a is determined uniquely.

Then we represent the ℓ-simplex c by the ℓ-tuple $\alpha = (a_1, a_2, \dots, a_\ell)$, where we write a_i instead of $a(i)$ for typographic reasons. The boundary operators then work as follows:

$$\begin{aligned} \partial_0 \alpha &= (a_2 - a_1, a_3 - a_1, \dots, a_\ell - a_1), \\ \partial_i \alpha &= (a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_\ell), \quad i = 1, 2, \dots, \ell. \end{aligned}$$

The degeneracy operator s_0 prepends 0 to the beginning of the sequence, and for $i \geq 1$, s_i duplicates the ith term. An analogous representation is used for the simplices of $K(\mathbb{Z}/m, 1)$.

Now if $\alpha = (a_1, \dots, a_\ell) \in K(\mathbb{Z}, 1)_\ell$ and $\beta = (b_1, \dots, b_\ell) \in K(\mathbb{Z}/m, 1)_\ell$ are simplices represented in this way, the desired simplicial isomorphism $\varphi\colon K(\mathbb{Z}, 1) \times_\tau K(\mathbb{Z}/m, 1) \to K(\mathbb{Z}, 1)$ is defined by

$$\varphi_\ell(\alpha, \beta) := (ma_1 + \iota(b_1), \dots, ma_\ell + \iota(b_\ell)),$$

where $\iota\colon \mathbb{Z}/m \to \mathbb{Z}$ is the identification of $\mathbb{Z}/m$ with $\{0, 1, \dots, m\} \subseteq \mathbb{Z}$. It is clear that φ_ℓ is a bijection between the sets of ℓ-simplices, and that both φ and φ^{-1} are polynomial-time computable.

We recall that in the twisted product $K(\mathbb{Z}, 1) \times_\tau K(\mathbb{Z}/m, 1)$ we have $s_i(\alpha, \beta) = (s_i\alpha, s_i\beta)$ for all i, and $\partial_i(\alpha, \beta) = (\partial_i\alpha, \partial_i\beta)$ for all $i \geq 1$. It is then straightforward to check that the mapping φ commutes with $s_0, \dots, s_\ell$ and with $\partial_1, \dots, \partial_\ell$.

The face operator ∂_0 is twisted, i.e., $\partial_0(\alpha, \beta) = (\tau(\beta) + \partial_0\alpha, \partial_0\beta)$ (here we write the group operation additively, unlike in the general discussion of twisted products earlier). From the requirement that φ commute with ∂_0, we can compute the appropriate twisting operator τ.

Namely, we have

$$\partial_0 \varphi_\ell(\alpha, \beta) = \Big(m(a_2 - a_1) + \iota(b_2) - \iota(b_1), \dots, m(a_\ell - a_1) + \iota(b_\ell) - \iota(b_1)\Big),$$

while

$$\varphi_{\ell-1}(\partial_0\alpha, \partial_0\beta) = \Big(m(a_2 - a_1) + \iota(b_2 - b_1), \dots, m(a_\ell - a_1) + \iota(b_\ell - b_1)\Big)$$

(where the subtraction in the argument of ι is in $\mathbb{Z}/m$, i.e., modulo m). It follows that τ has to be given by

$$\tau_\ell(\beta) = \Big(\iota(b_2) - \iota(b_1) - \iota(b_2 - b_1), \dots, \iota(b_\ell) - \iota(b_1) - \iota(b_\ell - b_1)\Big).$$

This is obviously a polynomial-time map, and a routine check of properties (i)–(iv) of a twisting operator in Definition 3.9 concludes the proof. $\square$

3.9 A pullback from a fibration of Eilenberg–MacLane spaces

For our construction of Postnikov systems, we will need an operation that is essentially a twisted Cartesian product, but in a somewhat different representation. We will have the following situation. We are given a simplicial set P, plus a simplicial mapping $f\colon P \to K(\pi, k+1)$, for some Abelian group π and a fixed $k \geq 1$.

Now we define a simplicial set Q as the *pullback* according to the following commutative diagram:

$$\begin{tikzcd} Q \arrow[r] \arrow[d] & E(\pi,k) \arrow[d, "\delta"] \\ P \arrow[r, "f"] & K(\pi,k+1) \end{tikzcd}$$

This means that Q is the simplicial subset of the Cartesian product $P \times E(\pi,k)$ consisting of the pairs (α, β) of simplices $\alpha \in P_\ell$, $\beta \in E(\pi,k)_\ell$ with $f(\alpha) = \delta(\beta)$.

As a simple consequence of Proposition 3.10 (twisted product) and of an explicit isomorphism of the pullback with a suitable twisted product, we obtain the following.

Corollary 3.18. *Given π, k, P, f as above, where π is a globally polynomial-time Abelian group, P is equipped with polynomial-time homology, and f is polynomial-time, all parameterized by the same parameter set $\mathcal{I}$, the pullback Q can be equipped with polynomial-time homology.*

Proof. Let τ be the twisting operator in the twisted product $K(\pi,k) \times_\tau K(\pi,k+1)$ at the end of Section 3.7, and let τ^* be the pullback of τ by f; that is, $\tau^*(\alpha) := \tau(f(\alpha))$. Then Proposition 3.10 yields polynomial-time homology for the twisted product $K(\pi,k) \times_{\tau^*} P$. According to [19, Prop. 18.7] (which is formulated in a more general setting), there is a simplicial isomorphism $\varphi\colon K(\pi,k) \times_{\tau^*} P \to Q$, given by

$$\varphi(\alpha,\beta) := (\psi(f(\alpha)) + \beta, \alpha),$$

where $\psi\colon K(\pi,k+1) \to E(\pi,k)$ is the *pseudo-section* given by

$$\psi(z)(i_0,\ldots,i_k) := z(0, i_0+1, \ldots, i_k+1),$$

with the same notation as in the definition of τ. Since both φ and its inverse are polynomial-time maps, we obtain polynomial-time homology for Q as needed.

In addition, setting $P = K(\pi,k+1)$, we have $Q = E(\pi,k)$ and we obtain the isomorphism $E(\pi,k) \cong K(\pi,k) \times_\tau K(\pi,k+1)$ mentioned at the end of Section 3.7. □

4 Postnikov systems

Let Y be a topological space, which we will assume to be given as a simplicial set equipped with polynomial-time homology. Moreover, we assume that Y is 1-connected. This is needed for the proof of correctness of the algorithm; the algorithm itself does not make use of any certificate of 1-connectedness, and in particular, we do not assume Y 1-reduced.

For our purposes, we define a *(simplicial) Postnikov system* of Y as the collection of simplicial sets and simplicial maps organized into the following commutative diagram,

$$\begin{tikzcd}
 & \vdots \arrow[d, "p_3"] \\
 & P_2 \arrow[d, "p_2"] \\
 & P_1 \arrow[d, "p_1"] \\
Y \arrow[ruu, "\varphi_2"] \arrow[ru, "\varphi_1"] \arrow[r, "\varphi_0"] & P_0 = *
\end{tikzcd}$$

where P_0 is a single point, and the following conditions hold:

(i) For each $k \geq 0$, the map $\varphi_k : Y \to P_k$ induces isomorphisms $\varphi_{k*} : \pi_i(Y) \to \pi_i(P_k)$ of homotopy groups for $0 \leq i \leq k$, while $\pi_i(P_k) = 0$ for $i \geq k+1$.

(ii) Each P_k, $k \geq 1$, is the pullback according to the following diagram (as in Section 3.9) for some map $\mathbf{k}_{k-1}\colon P_{k-1} \to K(\pi_k(Y), k+1)$:

$$\begin{tikzcd}
P_k \arrow[r] \arrow[d, "p_k"'] & E(\pi_k(Y), k) \arrow[d, "\delta"] \\
P_{k-1} \arrow[r, "\mathbf{k}_{k-1}"] & K(\pi_k(Y), k+1)
\end{tikzcd}$$

The simplicial sets $P_0, P_1, \ldots$ are called the *stages* of the Postnikov system, and the mappings $\mathbf{k}_i$ are called *Postnikov classes* (the terms *Postnikov factors* or *Postnikov invariants* are also used in the literature).

In the simplicial Postnikov system as introduced above, each P_k is a simplicial subset of the Cartesian product $P_{k-1} \times E(\pi_k(Y), k)$, and the map $p_k\colon P_k \to P_{k-1}$ is the projection to the first component.

In the rest of this section, we will prove Theorem 1.2. First we should make the statement precise.

Theorem 4.1 (Restatement of Theorem 1.2)**.** *Let $k \geq 2$ be fixed and let $(Y(I) : I \in \mathcal{I})$ be a simplicial set with polynomial-time homology, the main example being a finite simplicial complex, and let us suppose that Y is 1-connected (or simple; see the remark following Theorem 1.1). Then there is a polynomial-time algorithm that, given $I \in \mathcal{I}$, computes, for each $i \leq k$, the isomorphism type $\mathbf{m}_i = \mathbf{m}_i(I)$ of the homotopy group $\pi_i(Y(I))$. Furthermore, we can construct the following objects (i.e., write down the algorithms for the black boxes representing them, which use the black boxes defining Y as subroutines).*

- *Simplicial sets $P_0, P_1, \ldots, P_k$ with polynomial-time homology.*
- *Polynomial-time simplicial maps $\varphi_i\colon Y \to P_i$, $i \leq k$.*

- *Polynomial-time simplicial maps* $\mathbf{k}_{i-1}\colon P_{i-1} \to K(\pi_i, i+1)$, $i \le k$, *where we use the notation* $\pi_i := \mathrm{Ab}(\mathbf{m}_i)$ *for the canonical representation of the Abelian group described by* $\mathbf{m}_i$ *(see the text following Definition 2.3).*

All of these objects are parameterized by $\mathcal{I}$. *The* $P_i(I)$, *the* $(\varphi_i)_I$, *and the* $(\mathbf{k}_{i-1})_I$ *form a Postnikov system of* $Y(I)$.

4.1 The algorithm

Representing a simplicial map by an effective cocycle. In the Postnikov system algorithm, we will encounter the following situation. We consider a simplicial set $(U(I) : I \in \mathcal{I})$ with polynomial-time homology; let us write EC_*^U for the globally polynomial-time chain complex used in the polynomial-time homology, i.e., the one for which $C_*(U) \overset{\mathrm{P}}{\Longleftrightarrow} EC_*^U$.

Let us also consider a $(k+1)$-cocycle $\psi^{\mathrm{ef}} \in Z^{k+1}(EC_*^U; \pi)$ for some globally polynomial-time Abelian group π, also parameterized by $\mathcal{I}$; here the superscript "ef" should suggest that the cocycle belongs to the "effective" chain complex EC_*^U associated to U. Then ψ^{ef} can be represented by a finite matrix, since it is a homomorphism from the chain group EC_{k+1}^U of finite rank into π.

Now the strong equivalence $C_*(U) \overset{\mathrm{P}}{\Longleftrightarrow} EC_*^U$ defines, in particular, a chain map $f\colon C_*(U) \to EC_*^U$. We define a cocycle $\psi \in Z^{k+1}(C_*(U))$ as $\psi = f\psi^{\mathrm{ef}}$. As was discussed in Section 3.7, such a ψ canonically defines a simplicial map $\hat{\psi}\colon U \to K(\pi, k+1)$.

The point we want to make here is that $\hat{\psi}$ can be regarded as a polynomial-time simplicial map parameterized by pairs (I, ψ^{ef}).

Re-parameterizing the Postnikov system. In Theorem 4.1, we have the Postnikov system parameterized by the same parameter set $\mathcal{I}$ as the input simplicial set Y. This simplifies the formulation, but as we have already remarked earlier, it is not very efficient for an implementation, since it stipulates re-computing everything from scratch every time we call one of the black boxes representing the Postnikov system.

We are going to organize the algorithm somewhat differently. We are going to define a new parameter set $\mathcal{J}_k$, whose elements have the form $(I, F_k(I))$, where F_k is a polynomial-time mapping described below. The computation of $F_k(I)$ corresponds to a preprocessing, or "construction" of the Postnikov system. Then we will have the Postnikov system parameterized by $\mathcal{J}_k$ instead of $\mathcal{I}$, and this will allow for much more effective black boxes. This point of view is also very natural for presentation of the Postnikov system algorithm.

What kind of data should be included in $\mathcal{J}_k$ to describe the Postnikov system? First, given $I \in \mathcal{I}$, we need the homotopy groups $\pi_i(Y(I))$, $i \le k$. As in Theorem 4.1, we are going to represent each $\pi_i(Y(I))$ by its isomorphism type $\mathbf{m}_i$, and we use the notation $\pi_i = \mathrm{Ab}(\mathbf{m}_i)$. Thus $\mathbf{m}_1, \ldots, \mathbf{m}_k$ are included in $F_k(I)$.

Next, the Postnikov stage P_k is a simplicial subset of the product

$$P_k \subseteq E(\pi_1, 1) \times \cdots \times E(\pi_k, k),$$

and for describing it, we need the Postnikov classes $\mathbf{k}_{i-1}$, $i \le k$. We are going to have $\mathbf{k}_{i-1}$ represented by a cocycle $\kappa_{i-1}^{\mathrm{ef}} \in Z^{i+1}(EC_*^{P_{i-1}}; \pi_i)$, in the way described above, and $\kappa_1^{\mathrm{ef}}, \ldots, \kappa_{k-1}^{\mathrm{ef}}$ are also a part of $F_k(I)$.

This, of course, assumes that P_{i-1} has already been equipped with polynomial-time homology; indeed, the algorithm will proceed inductively, constructing P_{i-1} first, then $\kappa_{i-1}^{\mathrm{ef}}$ (and

thus $\mathbf{k}_{i-1}$), and then P_i. Here P_i with polynomial-time homology is obtained as the pullback as in the definition of a Postnikov system, using Corollary 3.18.

Finally, to describe the maps $\varphi_1, \dots, \varphi_k$, we need even more data. Namely, φ_k is, in particular, a simplicial map into $E(\pi_1, 1) \times \cdots \times E(\pi_k, k)$, and so we can write it as $(\ell_1, \dots, \ell_k)$, where ℓ_i goes into $E(\pi_i, i)$. Each ℓ_i is going to be specified using a cochain $\lambda_i^{\mathrm{ef}} \in Z^i(EC_*^Y; \pi_i)$. The construction of ℓ_i from λ_i^{ef} is described in the algorithm below; it is roughly similar to the construction of $\mathbf{k}_i$ from κ_i^{ef}, but there is a subtlety involved.

Hence the parameter $J \in \mathcal{J}_k$ describing the first k stages of the Postnikov system has the form

$$J = (I, \mathbf{m}_1, \lambda_1^{\mathrm{ef}}, \kappa_1^{\mathrm{ef}}, \mathbf{m}_2, \dots, \kappa_{k-1}^{\mathrm{ef}}, \mathbf{m}_k, \lambda_k^{\mathrm{ef}}).$$

Of course, the κ_i^{ef} and λ_i^{ef} have to satisfy certain consistency requirements, so that they describe a valid Postnikov system (up to stage k). These will be formulated and proved later.

The Postnikov system algorithm. Now we describe the way of computing $F_k(I)$, i.e., obtaining the values of $\mathbf{m}_1, \lambda_1^{\mathrm{ef}}, \kappa_1^{\mathrm{ef}}, \dots, \kappa_{k-1}^{\mathrm{ef}}, \mathbf{m}_k, \lambda_k^{\mathrm{ef}}$ from I (using the black boxes specifying Y, of course).

As was mentioned above, we proceed by induction. By definition, there is nothing to compute for $k = 0$ and, in order to make the induction start, we define P_0 to be a single point and φ_0 to be the constant map. Next, we assume that the algorithm for F_{k-1}, computing the parameters, is given and we are required to compute the components $\kappa_{k-1}^{\mathrm{ef}}$, $\mathbf{m}_k$, and λ_k^{ef}.

1. Construct the algebraic mapping cone $M_* := \mathrm{Cone}_*((\varphi_{k-1})_*)$, where $(\varphi_{k-1})_* \colon C_*(Y) \to C_*(P_{k-1})$ is the chain map induced by φ_{k-1}, as a chain complex with polynomial-time homology, by Proposition 3.8. By the proof of that proposition, the corresponding globally polynomial-time chain complex EC_*^M has $EC_{k+1}^M = EC_k^Y \oplus EC_{k+1}^{P_{k-1}}$.

2. Compute the homology group $H_{k+1}(EC_*^M)$ as a globally polynomial-time Abelian group. We let $\mathbf{m}_k$ be its isomorphism type, and let $\pi_k = \mathrm{Ab}(\mathbf{m}_k)$. We also have an explicit, polynomial-time isomorphism $H_{k+1}(EC_*^M) \cong \pi_k$, as in the definition of a globally polynomial-time Abelian group.

3. Choose a decomposition of the chain group EC_{k+1}^M of the form $EC_{k+1}^M = EZ_{k+1}^M \oplus \widetilde{EC}_{k+1}^M$, where EZ_{k+1}^M is the subgroup of all cycles, and $\widetilde{EC}_{k+1}^M$ is an arbitrary direct complement. Let $\rho\colon EC_{k+1}^M \to \pi_k$ be given as the projection
$$\rho\colon EC_{k+1}^M = EZ_{k+1}^M \oplus \widetilde{EC}_{k+1}^M \to EZ_{k+1}^M \to H_{k+1}(EC_*^M) \xrightarrow{\cong} \pi_k.$$
In other words, every chain $c \in EC_{k+1}^M$ has a unique expression as $c = z + \tilde{c}$, $z \in EZ_{k+1}^M$, $\tilde{c} \in \widetilde{EC}_k^M$, and $\rho(c)$ is the element of π_k corresponding to the homology class $[z] \in H_{k+1}(EC_*^M) \cong \pi_k$.

4. Using the decomposition of EC_{k+1}^M as in Step 1, we denote the restriction of ρ to EC_k^Y by λ_k^{ef} and the restriction to $EC_{k+1}^{P_{k-1}}$ by $\kappa_{k-1}^{\mathrm{ef}}$. In effect, to give ρ is the same as to give its two components λ_k^{ef} and $\kappa_{k-1}^{\mathrm{ef}}$.

5. In the strong equivalence $M_* \overset{\mathrm{P}}{\Longleftarrow\!\!\Longrightarrow} EC_*^M$, let f denote the composite chain map $M_* \to EC_*^M$. Then we obtain a cochain $\rho f\colon M_{k+1} \to \pi_k$. Again we have a direct sum decomposition $M_{k+1} = C_k(Y) \oplus C_{k+1}(P_{k-1})$. We define $\lambda_k\colon C_k(Y) \to \pi_k$ as the restriction of

ρf to the summand $C_k(Y)$ and $\ell_k : Y \to E(\pi_k, k)$ as the corresponding simplicial map; it is clearly polynomial-time.

It is easy to see that all the computations can be implemented in polynomial time. Perhaps only the decomposition in Step 3 may need some comment. The computation of EZ^M_{k+1} is a part of computing the homology group $H_{k+1}(EC^M_*)$. Then, given a basis of EZ^M_{k+1}, it suffices to extend it to a basis of the free Abelian group EC^M_{k+1}, which is also straightforward using the Smith normal form.

To prove correctness, we will need to verify that $\pi_k \cong \pi_k(Y)$, that $\kappa^{\mathrm{ef}}_{k-1}$ is a cocycle, that the image of the induced map $\varphi_k = (\varphi_{k-1}, \ell_k)$ lies in P_k, and that it satisfies the conditions in the definition of a Postnikov system. The proofs of all these claims are postponed to Section 4.3.

Remark: non-uniqueness. A Postnikov system of a space Y is typically not unique. The algorithm above involves some arbitrary choices, namely, the choice of the direct complement of EZ^M_{k+1} in Step 3, as well as the choice of the isomorphism of $H_{k+1}(EC^M_*)$ with $\mathrm{Ab}(\mathbf{m}_i)$. Performing these choices differently may result in a different Postnikov system.

At the same time, in an algorithm that uses a Postnikov system, such as the one in Corollary 1.3, we make many calls to the black boxes representing the Postnikov system, and we thus need that each time they refer to the same Postnikov system, for otherwise, the algorithm may not work correctly. This requirement is reflected in the definition of a parameterized simplicial set $(X(I) : I \in \mathcal{I})$, where I determines $X(I)$ uniquely.

One way of satisfying this requirement is to use only deterministic algorithms (no randomization). Then, although the algorithm makes some "arbitrary" choices, these choices are always made in the same way for a given input.

Another, more conceptual and practical way, is the re-parameterization as above: the results of all of the arbitrary choices are encoded in $F_k(I)$, and then the Postnikov stages $P_i(J)$ are defined uniquely, and similarly for the $(\mathbf{k}_i)_J$ and $(\varphi_i)_J$. In this case the computation of $F_k(I)$ may use randomized algorithms as well, which may be useful, e.g., for a fast computation of the Smith normal form.

4.2 Further properties of Eilenberg–MacLane spaces

Here we prepare several lemmas needed in the proof of correctness of our algorithm for computing Postnikov systems. The proofs are routine, but we have no good reference for these facts. Here, π will stand for an Abelian group.

We recall that $\mathrm{ev}\colon K(\pi, k)_k = E(\pi, k)_k \to \pi$ is the mapping assigning to each π-valued cocycle $z \in Z^k(\Delta^k; \pi)$ its value on the unique k-face of Δ^k. We can extend ev linearly to a homomorphism $\mathrm{ev}\colon C_k(K(\pi, k)) \to \pi$.

The first lemma is essentially just re-phrasing of the considerations in Section 3.7 concerning the correspondence of simplicial maps into $E(\pi, k)$ with cochains.

Lemma 4.2 (Lemma 24.2 in [19]). *Let $f\colon X \to E(\pi, k)$ be a simplicial map. Then the cochain $\kappa\colon C_k(X) \to \pi$ corresponding to it can be expressed as $\kappa = \mathrm{ev}\, f_*$, where $f_*\colon C_*(X) \to C_*(E(\pi, k))$ is the chain map induced by f.*

Also see [19, Lemma 24.3] for the corresponding statement for $K(\pi, k)$.

The next two lemmas deal with maps induced by ev in homology.

Lemma 4.3. *The homomorphism* $\mathrm{ev}\colon C_k(K(\pi,k)) \to \pi$ *induces an isomorphism* $H_k(K(\pi,k)) \to \pi$.

Proof. First we note that $C_k(K(\pi,k)) = Z_k(K(\pi,k))$, since $K(\pi,k)_{k-1} = \{0\}$. Then ev is easily seen to be surjective, and so it remains to prove that $\ker(\mathrm{ev}) = B_k(K(\pi,k))$.

Let us consider $z \in K(\pi,k)_{k+1} = Z^k(\Delta^{k+1};\pi)$; thus, z is given by the $(k+2)$-tuple $(g_0,\ldots,g_{k+1})$, where g_i is the value of z on $\partial_i\Delta^{k+1}$, and the cocycle condition reads $\sum_{i=0}^{k+1}(-1)^i g_i = 0$ (in π). On the other hand, considering z as a chain in $C_{k+1}(K(\pi,k))$, we have $dz = \sum_{i=0}^{k+1}(-1)^i\partial_i z$, and $\partial_i z$ is the k-cochain on Δ^k with value g_i (if $g_i = 0$, the term $\partial_i z$ is ignored in dz). Thus $\mathrm{ev}(dz) = \sum_{i=0}^{k+1}(-1)^i g_i = 0$, and so $B_k(K(\pi,k)) \subseteq \ker(\mathrm{ev})$.

For the reverse inclusion, we recall that there is a one-to-one correspondence, given by the mapping ev, between the nondegenerate k-simplices of $K(\pi,k)$ and the nonzero elements of π. Let us write σ_g for the unique k-simplex of $K(\pi,k)$ with $\mathrm{ev}\,\sigma_g = g$. Then a k-chain $c \in C_k(K(\pi,k))$ can be written as $c = \sum_{g\in\pi\setminus\{0\}} \alpha_g\cdot\sigma_g$, with finitely many nonzero coefficients α_g. We have $\mathrm{ev}(c) = 0$ iff $\sum_{g\in\pi\setminus\{0\}} \alpha_g g = 0$ in π.

By the above description of generators of $B_k(K(\pi,k))$, and since $k \geq 1$, we get that for every $g_1, g_2 \in \pi$, the chain $1\cdot\sigma_{g_1} + 1\cdot\sigma_{g_2}$ is homologous to $1\cdot\sigma_{g_1+g_2}$ (where terms involving σ_0 are to be ignored). Then by induction we get that a general chain $c = \sum_{g\in\pi\setminus\{0\}} \alpha_g\cdot\sigma_g$ is homologous to $1\cdot\sigma_s$, where $s = \sum_{g\in\pi\setminus\{0\}} \alpha_g g$. In particular, if $\mathrm{ev}\,c = 0$, then c is homologous to the zero chain, and so $c \in B_k(K(\pi,k))$ as claimed. □

Lemma 4.4. *The homomorphism*

$$h := \mathrm{ev} + \mathrm{ev}\colon\ \mathrm{Cone}_{k+1}(\delta_*) = C_k(E(\pi,k)) \oplus C_{k+1}(K(\pi,k+1)) \to \pi$$

sending (σ,τ) *to* $\mathrm{ev}\,\sigma + \mathrm{ev}\,\tau$ *induces an isomorphism* $H_{k+1}(\mathrm{Cone}_*(\delta_*)) \to \pi$.

Proof. For brevity, we write $E = E(\pi,k)$ and $K = K(\pi,k+1)$ since there are no other Eilenberg–MacLane spaces in this proof.

In order to claim that h induces a map in homology, we verify that it vanishes on all boundaries. Thus, let $(\sigma',\tau') \in \mathrm{Cone}_{k+2}(\delta_*)$ be a generator, $\sigma' \in E_{k+1}$, $\tau' \in K_{k+2}$. According to the formula (1) in Section 3.3 we have $d^{\mathrm{Cone}_*}(\sigma',\tau') = (-d^E\sigma', \delta_*(\sigma') + d^K\tau')$. Since τ' is a cocycle, we have $\mathrm{ev}(d^K\tau') = 0$, as we saw in the proof of Lemma 4.3. Moreover, it is easily checked that $\mathrm{ev}(d^E\sigma') = \mathrm{ev}(\delta_*(\sigma'))$. It follows that h indeed vanishes on boundaries and induces a homomorphism $h_*\colon H_{k+1}(\mathrm{Cone}_*(\delta_*)) \to \pi$.

Now we consider the canonical inclusion $C_*(K) \to \mathrm{Cone}_*(\delta_*)$, which is a chain map, and thus it induces a map in homology, as in the following diagram (here we use that $C_{k+1}(K) = Z_{k+1}(K)$ and $C_k(E) = Z_k(E)$):

$$\begin{tikzcd}
 & C_{k+1}(K) \arrow[r, "i"] \arrow[d] & C_k(E) \oplus C_{k+1}(K) = \mathrm{Cone}_{k+1}(\delta_*) \arrow[d] \\
\pi & H_{k+1}(K) \arrow[l, "\mathrm{ev}_*", "\cong"'] \arrow[r, "i_*", "\cong"'] & H_{k+1}(\mathrm{Cone}_*(\delta_*))
\end{tikzcd}$$

Here ev_* on the left in the bottom row is the isomorphism induced by ev as in Lemma 4.3.

The map i_* is an isomorphism by the long exact homology sequence of the pair $(\mathrm{Cone}(\delta_*), C_*(K))$, because the quotient $\mathrm{Cone}(\delta_*)/C_*(K) \cong C_*(E)^{\uparrow}$ is the shift of the chain complex of a contractible simplicial set E (see e.g. [19, Proposition 21.5, Theorem 23.10]), and thus all homology groups of this quotient vanish except for the one in dimension 1.

Finally, it suffices to verify that $h_* i_* = \mathrm{ev}_*$, but this is clear, since the composition on the left maps $[\tau] \xmapsto{i_*} [(0,\tau)] \xmapsto{h_*} \mathrm{ev}\,\tau$. $\square$

4.3 Correctness of the algorithm

Here we provide a proof of correctness for the algorithm above. It uses more or less standard methods, but we do not know of an accessible presentation in the literature. In this part, we are going to use somewhat more advanced topological notions without defining them; we refer to standard textbooks, such as [15].

We assume the correctness of the algorithm for $k-1$. For brevity we write $\varphi = \varphi_{k-1}$, $P = P_{k-1}$, $K = K(\pi_k, k+1)$, and $E = E(\pi_k, k)$.

Checking $\boldsymbol{\pi_k \cong \pi_k(Y)}$. We recall that the algorithm sets up an isomorphism $H_{k+1}(EC_*^M) \cong \pi_k$; thus we need to verify that $H_{k+1}(EC_*^M) \cong \pi_k(Y)$. Let $\mathrm{Cyl}\,\varphi$ be the mapping cylinder of $\varphi\colon Y \to P$, i.e., the simplicial set $(Y \times \Delta^1 \cup P)/\sim$, where $\sim$ is the equivalence identifying $(y, 0)$ with $f(y)$, $y \in Y$. Let us also identify Y with $Y \times \{1\}$, the "top copy" of Y in $\mathrm{Cyl}\,\varphi$.

Using the Eilenberg–Zilber reduction, it is easy to check that the chain complex of the pair $(\mathrm{Cyl}\,\varphi, Y)$ has a reduction to $M_* = \mathrm{Cone}_*(\varphi_*)$. Hence

$$H_{k+1}(\mathrm{Cyl}\,\varphi, Y) \cong H_{k+1}(M_*) \cong H_{k+1}(EC_*^M).$$

Using the fact that $\mathrm{Cyl}\,\varphi$ is homotopy equivalent to P, and the assumption $\pi_i(P) = 0$ for $i \geq k$, the long exact sequence of homotopy groups for the pair $(\mathrm{Cyl}\,\varphi, Y)$ yields that this pair is k-connected and $\pi_k(Y) \cong \pi_{k+1}(\mathrm{Cyl}\,\varphi, Y)$. Due to the k-connectedness of $(\mathrm{Cyl}\,\varphi, Y)$, the Hurewicz isomorphism yields $\pi_{k+1}(\mathrm{Cyl}\,\varphi, Y) \cong H_{k+1}(\mathrm{Cyl}\,\varphi, Y)$. Putting all these isomorphisms together we obtain $\pi_k(Y) \cong \pi_k$, as desired.

The cochain $\boldsymbol{\kappa^{\mathrm{ef}}_{k-1}}$ is a cocycle. We recall that $\kappa^{\mathrm{ef}}_{k-1}$ is the composition

$$\kappa^{\mathrm{ef}}_{k-1}\colon EC^{P_{k-1}}_{k+1} \hookrightarrow EC^Y_k \oplus EC^{P_{k-1}}_{k+1} = EC^M_{k+1} \xrightarrow{\rho} \pi_k$$

The inclusion, being a chain map, preserves boundaries, and ρ, by definition, vanishes on them. Thus the composite $\kappa^{\mathrm{ef}}_{k-1}$ also vanishes on boundaries and is indeed a cocycle.

The map $\boldsymbol{\varphi_k}$ takes values in $\boldsymbol{P_k}$. First we will need a description of the cocycle κ_{k-1} similar to that of λ_k. Namely, the remark following the proof of Proposition 3.8 says that κ_{k-1} can be also obtained as a restriction of ρf from Step 5 to $C_{k+1}(P_{k-1})$.[26] Thus, denoting the inclusions of the two summands by $i\colon C_{k+1}(P_{k-1}) \to M_{k+1}$ and $j\colon C_k(Y) \to M_{k+1}$, we can write $\kappa_{k-1} = \rho f i$ and $\lambda_k = \rho f j$.

Now, we will verify that the map $\varphi_k = (\varphi, \ell_k)\colon Y \to P \times E$ has image in the pullback P_k, which amounts to showing that $\mathbf{k}_{k-1}\varphi = \delta \ell_k$. This will follow easily from the following equality of cochains in $C^{k+1}(Y; \pi_k)$:

$$\kappa_{k-1}\varphi_* = \lambda_k d^Y, \tag{13}$$

where $\varphi_*\colon C_{k+1}(Y) \to C_{k+1}(P)$ is the chain map induced by φ and d^Y is the differential in $C_*(Y)$. We have $\kappa_{k-1}\varphi_* - \lambda_k d^Y = \rho f(i\varphi_* - jd^Y)$. As above, ρf maps boundaries in

[26]On the other hand, λ_k in general cannot be computed solely from λ_k^{ef} and the effective homology of Y. A notable exception to this is when $C_*(Y) = EC_*^Y$, as happens e.g. for finite simplicial complexes. In this case we have $\lambda_k = \lambda_k^{\mathrm{ef}}$.

M_* to 0, so it suffices to show that the images of $i\varphi_* - jd^Y$ are boundaries—but by the formula for the differential in the algebraic mapping cone, we have that for every $\sigma \in Y_{k+1}$, $(i\varphi_* - jd^Y)(\sigma) = d^{M_*}(\sigma, 0)$ is indeed a boundary.

Using Lemma 4.2, we find that $\kappa_{k-1}\varphi_* = (\mathrm{ev}(\mathbf{k}_{k-1})_*)\varphi_* = \mathrm{ev}(\mathbf{k}_{k-1}\varphi)_*$. It is also easy to verify from the definitions that $\mathrm{ev}(\delta\ell_k)_* = \lambda_k d^Y$, and so the equality (13) of cochains yields the desired equality $\mathbf{k}_{k-1}\varphi = \delta\ell_k$ of simplicial maps.

The maps induced by φ_k in homotopy. Considering the long exact sequence of homotopy groups of the fibration $K(\pi_k, k) \to P_k \to P$ and using the assumption $\pi_i(P) = 0$ for $i \geq k$, it is straightforward to check that $\pi_i(P_k) = 0$ for $i \geq k+1$, and that the maps $\pi_i(Y) \to \pi_i(P_k)$ induced by φ_k are isomorphisms for $i \leq k-1$. For establishing condition (i) in the definition of a Postnikov system, it remains to verify that $(\varphi_k)_*\colon \pi_k(Y) \to \pi_k(P_k)$ is an isomorphism as well.

To this end, we begin with the diagram

$$\begin{tikzcd} Y \arrow[r, "\varphi_k"] \arrow[d, "\varphi"] & P_k \arrow[r] \arrow[d, "p_k"] & E \arrow[d, "\delta"] \\ P \arrow[r, equal] & P \arrow[r] & K, \end{tikzcd}$$

where the right square is the pullback diagram defining P_k. Next, we replace each of the spaces in the bottom row with the mapping cylinder of the respective vertical map, so that the vertical maps become inclusions (of the domain in the cylinder); the horizontal maps of the cylinders are then induced in a canonical way.

$$\begin{tikzcd} Y \arrow[r, "\varphi_k"] \arrow[d] & P_k \arrow[r] \arrow[d] & E \arrow[d] \\ \operatorname{Cyl}\varphi \arrow[r] & \operatorname{Cyl} p_k \arrow[r] & \operatorname{Cyl}\delta \end{tikzcd} \tag{14}$$

Lemma 4.5. *The map $\pi_{k+1}(\operatorname{Cyl}\varphi, Y) \to \pi_{k+1}(\operatorname{Cyl} p_k, P_k)$ induced by the left square of the last diagram is an isomorphism.*

We first finish the proof of correctness of the algorithm assuming the lemma. We consider the long exact sequences coming from the pairs $(\operatorname{Cyl}\varphi, Y)$ and $(\operatorname{Cyl} p_k, P_k)$:

$$\begin{tikzcd} 0 = \pi_{k+1}(\operatorname{Cyl}\varphi) \arrow[r] \arrow[d, "\cong"] & \pi_{k+1}(\operatorname{Cyl}\varphi, Y) \arrow[r] \arrow[d, "\cong"] & \pi_k(Y) \arrow[r] \arrow[d, "\varphi_{k*}"] & \pi_k(\operatorname{Cyl}\varphi) = 0 \arrow[d, "\cong"] \\ 0 = \pi_{k+1}(\operatorname{Cyl} p_k) \arrow[r] & \pi_{k+1}(\operatorname{Cyl} p_k, P_k) \arrow[r] & \pi_k(P_k) \arrow[r] & \pi_k(\operatorname{Cyl} p_k) = 0 \end{tikzcd}$$

The second vertical isomorphism is proved in the lemma and the other two follow from $\pi_i(P) = 0$ for $i \geq k$, since both of the cylinders deform onto the base P. Then the five-lemma implies that φ_{k*} is an isomorphism on π_k, which completes the proof of condition (i) from the definition of a Postnikov system. All that remains is to prove the lemma.

Proof of Lemma 4.5. We will show that both the right square and the composite square induce an isomorphism in the relative homotopy groups of the vertical pairs in dimension $k+1$. We start with the composite square.

Since both $(\operatorname{Cyl}\varphi, Y)$ and $(\operatorname{Cyl}\delta, E)$ are k-connected, it suffices to prove that the square induces an isomorphism on the $(k+1)$-st homology group. We use that the chain complexes of these pairs are isomorphic to the respective (reduced) algebraic mapping cones. We find that the chain map $C_*(\operatorname{Cyl}\varphi, Y) \to C_*(\operatorname{Cyl}\delta, E)$ is actually $\ell_{k*} \oplus \mathbf{k}_{(k-1)*}$; this can be seen using the diagram

$$\begin{tikzcd} Y \arrow[r, "\ell_k"] \arrow[d, "\varphi"'] & E \arrow[d, "\delta"] \\ P \arrow[r, "\mathbf{k}_{k-1}"] & K. \end{tikzcd}$$

Then we consider the diagram

$$\begin{tikzcd} C_k(Y) \oplus C_{k+1}(P) \arrow[rr, "\ell_{k*}\oplus\mathbf{k}_{(k-1)*}"] \arrow[dr, "\lambda_k+\kappa_{k-1}"'] & & C_k(E) \oplus C_{k+1}(K), \arrow[dl, "\mathrm{ev}+\mathrm{ev}"] \\ & \pi_k & \end{tikzcd}$$

which commutes in view of Lemma 4.2. The left map $\lambda_k + \kappa_{k-1}$ equals ρf, and since both ρ and f induce isomorphisms in homology, so does $\lambda_k + \kappa_{k-1}$. The map $\mathrm{ev} + \mathrm{ev}$ induces an isomorphism in homology by Lemma 4.4. Therefore the same is true for the horizontal map, and hence the composite square in the diagram (14) induces an isomorphism in the $(k+1)$st homotopy groups of the vertical pairs, as claimed.

It remains to study the right square. Before we passed to mapping cylinders, the original square was a pullback. The original vertical maps are fibrations, and consequently, the induced map on fibers (which are both $K(\pi_k, k)$) is an isomorphism. Next, there is an isomorphism $\pi_{k+1}(\operatorname{Cyl} p_k, P_k) \cong \pi_k(\operatorname{fib} p_k)$, and a similar one for δ. From their description below it will be apparent that this isomorphism is natural so that the square

$$\begin{tikzcd} \pi_{k+1}(\operatorname{Cyl} p_k, P_k) \arrow[r] \arrow[d, "\cong"'] & \pi_{k+1}(\operatorname{Cyl}\delta, E) \arrow[d, "\cong"] \\ \pi_k(\operatorname{fib} p_k) \arrow[r, "\cong"'] & \pi_k(\operatorname{fib}\delta). \end{tikzcd}$$

commutes. We will thus be able to conclude that $\pi_{k+1}(\operatorname{Cyl} p_k, P_k) \to \pi_{k+1}(\operatorname{Cyl}\delta, E)$ is indeed an isomorphism as required.

The required map $\pi_{k+1}(\operatorname{Cyl} p_k, P_k) \to \pi_k(\operatorname{fib} p_k)$ is defined via representatives. To this end, we represent an element of $\pi_{k+1}(\operatorname{Cyl} p_k, P_k)$ by a map $f\colon I^{k+1} \to \operatorname{Cyl} p_k$ that sends the face I^k (where the last coordinate is zero) to P_k and the union of the remaining faces, which we denote by J^k, to the basepoint (here I^{k+1} denotes the unit cube). Now composing f with the projection $\mathrm{pr}\colon \operatorname{Cyl} p_k \to P$ we obtain $g = \mathrm{pr} \circ f\colon I^{k+1} \to P$, which we lift along p_k to $\widetilde{g}\colon I^{k+1} \to P_k$. One may prescribe the values on all the faces except for one. Here we decide that $\widetilde{g}$ agrees with f on the only interesting face I^k and that it is constant onto the basepoint on the neighboring faces. Finally, the restriction to the remaining face (opposite to I^k) gives us a map $\widetilde{g}_1\colon I^k \to \operatorname{fib} p_k$, and this is the representative of the image of $[f]$ under the desired map $\pi_{k+1}(\operatorname{Cyl} p_k, P_k) \to \pi_k(\operatorname{fib} p_k)$.

It remains to show that this map is indeed an isomorphism. For this, we consider the following diagram, whose top row is the long exact sequence of the pair $(\operatorname{Cyl} p_k, P_k)$, and

whose bottom row is associated with the fibration p_k.

$$\begin{tikzcd}
\cdots \arrow[r] & \pi_{k+1}P_k \arrow[r] \arrow[d, "\mathrm{id}"] & \pi_{k+1}\operatorname{Cyl}p_k \arrow[r] \arrow[d, "\cong"] \arrow[dr, phantom, "\text{(A)}"] & \pi_{k+1}(\operatorname{Cyl}p_k, P_k) \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(B)}"] & \pi_k P_k \arrow[r] \arrow[d, "\mathrm{id}"] & \pi_k \operatorname{Cyl}p_k \arrow[r] \arrow[d, "\cong"] & \cdots \\
\cdots \arrow[r] & \pi_{k+1}P_k \arrow[r] & \pi_{k+1}P \arrow[r] & \pi_k \operatorname{fib}p_k \arrow[r] & \pi_k P_k \arrow[r] & \pi_k P \arrow[r] & \cdots
\end{tikzcd}$$

The isomorphism will follow from the five-lemma once we show that the squares (A) and (B) commute up to a sign. The square (A) anticommutes because the path through the bottom left corner consists of lifting g as above but with the restriction to J^k being constant onto the basepoint. One can obtain this by first flipping I^{k+1} along the last coordinate and then lifting as above. The flipping amounts to multiplication by -1 on $\pi_{k+1}(P)$. The square denoted by (B) commutes by an easy inspection: the map $\widetilde{g}_1$ is homotopic inside P_k with $f|_{I^k}$ (the image in the top right corner of that square), the required homotopy being $\widetilde{g}$. $\square$

5 The extension problem

Proof of Theorem 1.4. Here we prove the result about testing extendability of a map using tools from [6]. We are given simplicial sets $A \subseteq X$ and Y and a simplicial map $f\colon A \to Y$, where X is finite, $\dim X \le 2k-1$, and Y is $(k-1)$-connected.

First, by [42, Theorem 7.6.22], a continuous extension of f to X exists, under these assumptions, if and only if the composition $\varphi_{2k-2}f\colon A \to P_{2k-2}$ admits a continuous extension to X, where $\varphi_{2k-2}\colon Y \to P_{2k-2}$ is the map in the Postnikov system of Y. By the homotopy extension property, this happens precisely when there exists a map $X \to P_{2k-2}$, whose restriction to A is homotopic to $\varphi_{2k-2}f$. In terms of homotopy classes of maps, this is if and only $[\varphi_{2k-2}f]$ lies in the image of the restriction map $\rho\colon [X, P_{2k-2}] \to [A, P_{2k-2}]$.

The algorithm in Corollary 1.3 for computing $[X, Y]$ actually computes $[X, P_{2k-2}]$. The isomorphism $[X, Y] \cong [X, P_{2k-2}]$ holds only for $\dim X \le 2k-2$, but the computation of $[X, P_{2k-2}]$ works correctly for X of arbitrary dimension. Thus, in the setting of Theorem 1.4, we can compute the Abelian group $[X, P_{2k-2}]$ represented by generators, which are specified as simplicial maps[27] $X \to P_{2k-2}$, and relations (it is *fully effective* in the terminology of [6]).

For the simplicial subset $A \subseteq X$, we similarly compute $[A, P_{2k-2}]$. As we recall from [6], the group operation in $[X, P_{2k-2}]$ is induced by an operation $\boxplus$ on $\operatorname{SMap}(X, P_{2k-2})$, which is defined simplexwise (i.e., $(f \boxplus g)(\sigma) = f(\sigma) \boxplus g(\sigma)$). This easily implies that the restriction map ρ is a group homomorphism.

Given an element (homotopy class) $[g] \in [X, P_{2k-2}]$, represented by a simplicial map g, we consider the restriction $g|_A$ as a representative of an element of $[A, P_{2k-2}]$, and we can express it using the generators of $[A, P_{2k-2}]$. Thus, ρ is polynomial-time computable, and we can compute the image $\operatorname{im}\rho$ as a subgroup of $[A, P_{2k-2}]$ (by computing the images of the generators of $[X, P_{2k-2}]$ and the subgroup generated by them).

Then, given a simplicial map $f\colon A \to Y$, we compute the corresponding element $[\varphi_{2k-2}f] \in [A, P_{2k-2}]$ and test (in polynomial time) whether it lies in the image of ρ. This is the desired algorithm for testing the extendability of f.

In case $\dim X \le 2k-2$ we have $[X, Y] \cong [X, P_{2k-2}]$ and $[A, Y] \cong [A, P_{2k-2}]$. Thus, if $x \in \operatorname{im}\rho$, we can compute the preimage $\rho^{-1}(x)$ as a coset in $[X, P_{2k-2}]$ (since we have ρ

[27] Actually, a more compact *cochain representation* is used in [6], but for our purposes, we can think of explicit simplicial maps.

represented by a matrix), and this coset is isomorphic to $[X, Y]_f$ as needed. This concludes the proof. $\square$

Acknowledgments

We would like to thank Francis Sergeraert for many useful discussion and extensive advice; in particular, the current proof of Lemma 3.17 is mostly due to him—not speaking of his leading role in the development of effective homology, which constitutes the foundation of our algorithmic methods. We also thank Marek Filakovský for useful discussions.

References

[1] V. Álvarez, J.A. Armario, M.D. Frau, and P. Real. Algebra structures on the comparison of the reduced bar construction and the reduced W-construction. *Commun. Algebra*, 37(10):3643–3665, 2009.

[2] D. J. Anick. The computation of rational homotopy groups is $\#\wp$-hard. Computers in geometry and topology, Proc. Conf., Chicago/Ill. 1986, Lect. Notes Pure Appl. Math. 114, 1–56, 1989.

[3] D. W. Barnes and L. A. Lambe. A fixed point approach to homological perturbation theory. *Proc. AMS*, 112:881–892, 1991.

[4] E. H. Brown, Jr. Twisted tensor products. I. *Ann. Math. (2)*, 69:223–246, 1959.

[5] E. H. Brown (jun.). Finite computability of Postnikov complexes. *Ann. Math. (2)*, 65:1–20, 1957.

[6] M. Čadek, M. Krčál, J. Matoušek, F. Sergeraert, L. Vokřínek, and U. Wagner. Computing all maps into a sphere. Preprint, arXiv/1105.6257, 2011. Extended abstract in *Proc. ACM–SIAM Symposium on Discrete Algorithms* (SODA 2012).

[7] M. Čadek, M. Krčál, J. Matoušek, L. Vokřínek, and U. Wagner. Extendability of continuous maps is undecidable. Preprint, 2012.

[8] E. B. Curtis. Simplicial homotopy theory. *Advances in Math.*, 6:107–209, 1971.

[9] S. Eilenberg and S. Mac Lane. On the groups of $H(\Pi, n)$. I. *Ann. of Math. (2)*, 58:55–106, 1953.

[10] S. Eilenberg and S. Mac Lane. On the groups $H(\Pi, n)$. II. Methods of computation. *Ann. of Math. (2)*, 60:49–139, 1954.

[11] M. Filakovský. Effective chain complexes for twisted products. Preprint, arXiv: 1209.1240, 2012.

[12] G. Friedman. An elementary illustrated introduction to simplicial sets. Rocky Mountain J. Math., to appear. Preprint arXiv:math/0809.4221v3, 2011.

[13] P. G. Goerss and J. F. Jardine. *Simplicial homotopy theory.* Birkhäuser, Basel, 1999.

[14] R. Gonzalez-Diaz and P. Real. Simplification techniques for maps in simplicial topology. *J. Symb. Comput.*, 40:1208–1224, October 2005.

[15] A. Hatcher. *Algebraic Topology.* Cambridge University Press, Cambridge, 2001. Electronic version available at `http://math.cornell.edu/hatcher#AT1`.

[16] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Computing*, 8:499–507, 1981.

[17] S. O. Kochman. *Stable homotopy groups of spheres. A computer-assisted approach.* Lecture Notes in Mathematics 1423. Springer-Verlag, Berlin etc., 1990.

[18] M. Krčál, J. Matoušek, and F. Sergeraert. Polynomial-time homology for simplicial Eilenberg–MacLane spaces. Preprint, arXiv/1201.6222, 2011.

[19] J. P. May. *Simplicial objects in algebraic topology.* Chicago Lectures in Mathematics. University of Chicago Press, Chicago, IL, 1992. Reprint of the 1967 original; the page numbers do not quite agree with the 1967 edition.

[20] J. C. Moore. Homological algebra and the cohomology of classifying spaces (in French). *Séminaire H. Cartan 1959/60*, exp. 7, 1959.

[21] J. R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, Reading, MA, 1984.

[22] R. Niedermeier. *Invitation to fixed parameter algorithms.* Oxford Lecture Series in Mathematics and its Applications 31. Oxford University Press, 2006.

[23] P. S. Novikov. Ob algoritmičeskoĭ nerazrešimosti problemy toždestva slov v teorii grupp (on the algorithmic unsolvability of the word problem in group theory). *Trudy Mat. Inst. im. Steklova*, 44:1–143, 1955.

[24] D. C. Ravenel. *Complex Cobordism and Stable Homotopy Groups of Spheres (2nd ed.).* Amer. Math. Soc., 2004.

[25] P. Real. Algorithms for computing effective homology of classifying spaces (in Spanish). PhD. Thesis, Facultad de Mathemáticas, Univ. de Sevilla, 1993. Available online at `http://fondosdigitales.us.es/media/thesis/1426/C_043-139.pdf`.

[26] P. Real. An algorithm computing homotopy groups. *Mathematics and Computers in Simulation*, 42:461—465, 1996.

[27] P. Real. Homological perturbation theory and associativity. *Homology Homotopy Appl.*, 2:51–88, 2000.

[28] A. Romero, G. Ellis, and J. Rubio. Interoperating between computer algebra systems: computing homology of groups with Kenzo and GAP. In *Proc. ISAAC, ACM, New York*, pages 303–310, 2009. Available on-line at `http://hamilton.nuigalway.ie/preprints/sigproc-sp.rev1.pdf`.

[29] A. Romero, J. Rubio, and F. Sergeraert. Computing spectral sequences. *J. Symb. Comput.*, 41(10):1059–1079, 2006.

[30] A. Romero and F. Sergeraert. Effective homotopy of fibrations. *Applicable Algebra in Engineering, Communication and Computing*, 23(1-2):85–100, 2012.

[31] B. H. Roune and E. Sáenz de Cabezón. Complexity and algorithms for Euler characteristic of simplicial complexes. Preprint arXiv:1112.4523, `http://arxiv.org/pdf/1112.4523v1`, 2011.

[32] J. Rubio. Effective homology of iterated loop spaces (in French). PhD. Thesis, Univ. Grenoble I, 1991.

[33] J. Rubio and F. Sergeraert. Constructive algebraic topology. *Bull. Sci. Math.*, 126(5):389–412, 2002.

[34] J. Rubio and F. Sergeraert. Algebraic models for homotopy types. *Homology, Homotopy and Applications*, 17:139–160, 2005.

[35] J. Rubio and F. Sergeraert. Constructive homological algebra and applications. Preprint, arXiv:1208.3816, 2012. Written in 2006 for a MAP Summer School at the University of Genova.

[36] J. Rubio and F. Sergeraert. Homology of iterated loop spaces. Manuscript in preparation, 2012.

[37] R. Schön. Effective algebraic topology. *Mem. Am. Math. Soc.*, 451:63 p., 1991.

[38] F. Sergeraert. The computability problem in algebraic topology. *Adv. Math.*, 104(1):1–29, 1994.

[39] F. Sergeraert. Introduction to combinatorial homotopy theory. Available at `http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/`, 2008.

[40] Weishu Shih. Homologie des espaces fibres. *Publ. Math. de l'IHES*, 13:93–176, 1962.

[41] J. R. Smith. m-structures determine integral homotopy type. Preprint, arXiv:math/9809151v1, 1998.

[42] E. H. Spanier. *Algebraic topology.* McGraw Hill, 1966.

[43] N. E. Steenrod. Cohomology operations, and obstructions to extending continuous functions. *Advances in Math.*, 8:371–416, 1972.

[44] A. Storjohann. Near optimal algorithms for computing Smith normal forms of integer matrices. In *International Symposium on Symbolic and Algebraic Computation*, pages 267–274, 1996.